

EasyRent: Alquiler de ayudas técnicas

Grado en Ingeniería Multimedia



Trabajo Fin de Grado

Autor:
Carlos García Gómez

Tutor/es:
José Vicente Berná Martínez

Mayo 2019



Universitat d'Alacant
Universidad de Alicante

Licencia

Se permite la reproducción, distribución y comunicación pública de la obra, incluso con fines comerciales siempre y cuando reconozca y cite la obra de la forma especificada por el autor o el licenciente.



Resumen

En el presente Proyecto de Fin de Grado se ha desarrollado una plataforma para facilitar y poner en contacto a cualquier persona que necesite hacer uso de alguna ayuda técnica con las empresas que distribuyen estos productos. Esta plataforma además de poner en contacto a la oferta y la demanda permite alquilar o reservar un producto a través de ella.

Por otro lado, EasyRent dispone de un pequeño backoffice que les permite a las empresas dadas de alta gestionar los productos que van a ofertar en la plataforma.

La idea de realizar un proyecto de estas características surgió debido a que, tuve la necesidad de alquilar un andador para un familiar muy cercano y fue una auténtica odisea y viví en mis carnes la problemática detallada en esta memoria, ya que por ejemplo este tipo de productos carecen de visibilidad y difusión.

El desarrollo del proyecto ha seguido los principios de las metodologías ágiles definiendo sprints de aproximadamente 15 días en los que se debía llevar a cabo una serie de tareas que debían concluir con un “producto” visible.

En cuanto a la implementación del sistema después de realizar un pequeño análisis de las tecnologías actuales se han empleado las que a mi juicio eran las más apropiadas para este tipo de trabajo y para mi formación de cara al mundo laboral que me espera. Entre ellas destacaría *Angular*, *NodeJs*, servicios de terceros como Google Maps...

Por todo esto y atendiendo a los resultados obtenidos podemos afirmar que se han cumplido todos los requisitos establecido obteniendo un producto funcional y no muy distante de algo profesional, así que me siento orgulloso del trabajo realizado para satisfacer un problema muy presente actualmente, además de demostrar y poner en práctica todo lo aprendido durante mi formación cursando el grado en Ingeniería Multimedia por la Universidad de Alicante.

Motivación, justificación y objetivo general

Después de superar el cuarto curso de ingeniería multimedia basado en la metodología ABP le toca el turno al trabajo final, este trabajo me lo planteo como un reto personal, no solo para seguir poniendo en práctica todos los conocimientos adquiridos en estos cuatro años, sino que también para seguir aumentando mis habilidades.

Durante el pasado curso realice junto a mis compañeras una plataforma para los profesionales de la mutilación genital femenina con multitud de funcionalidades que resultaron ser de gran utilidad para este sector, también mejoramos nuestros conocimientos en planificación, organización e implementación de servicios web. Pero lo más importante es que fuimos capaces de transformar requerimientos y necesidades de unos profesionales en soluciones informáticas personalizadas y funcionales.

Supuso un gran desafío encontrar un tipo de proyecto que fuera capaz de incluir todo lo anterior, que fuese de utilidad y de interés para la sociedad o para una empresa y que por supuesto pueda motivarme y por tanto pueda incluirlo en mi portfolio de cara a un posible futuro laboral.

A raíz de un problema de movilidad debido a una caída de mi abuela, tuvimos la necesidad de hacer uso de un andador para que pudiera caminar de manera digna. El primer paso fue localizar establecimientos donde se pudiera adquirir uno de ellos, este proceso fue fácil, ya que, en los hospitales te recomiendan sitios donde comprarlos. Una vez en el establecimiento vimos que existían multitud de modelos diferentes los cuales se diferencian en una cantidad económica importante, pero mi abuela quería probarlos unos días para asegurarse que modelo se adaptaba mejor a sus necesidades, porque estos andadores tienen precios que pueden suponer una inversión importante a personas mayores que únicamente se sustentan con su pensión de jubilación.

Por esta razón decidimos buscar opciones para alquilar estos dispositivos para poder probarlos libremente sin ningún compromiso de compra. Aquí fue donde empezó el verdadero quebradero de cabeza, ya que, localizar empresas de alquiler de andadores o pequeños vehículos/scooters para personas para movilidad reducida en Benidorm es bastante complicado debido a que estas empresas mayoritariamente se sustentan con acuerdos con hospitales, residencias para la tercera edad o incluso hoteles de la ciudad a causa de la popularidad de estos vehículos entre turistas extranjeros y no consideran importante darse a conocer a más personas que puedan necesitar hacer uso de este tipo de vehículos y así ampliar por ejemplo el número

de clientes. Además, resulto bastante complejo el simple hecho de poder comprar precios entre las diferentes empresas de alquiler, ya que, cada una aplica sus tarifas y sus normas especiales.

Por tanto, mi objetivo principal es lograr poner en contacto a las empresas de alquiler de aparatos para facilitar la movilidad con cualquier persona que pueda necesitar hacer uso de ellos por cualquier circunstancia, esto podría lograrse con el típico sistema donde las empresas colocan la información que desean dar a conocer para que los potenciales clientes la encuentren más rápidamente.

Por otro lado, como objetivo general de este trabajo es el diseño y desarrollo de un servicio web multiempresa anteriormente mencionado donde las empresas van a ofertar los productos disponibles para el alquiler añadiendo además información relativa al producto y a la empresa y los clientes podrán acceder a este sistema para buscar la información que deseen.

Por esta razón y habiendo sufrido las consecuencias de este proceso costoso de alquilar un andador he encontrado la motivación y un objetivo muy claro para realizar este proyecto final que no es otra que es poner a prueba mis habilidades y aplicar los conocimientos de ingeniería adquiridos durante estos cuatro años para poder evitar que otras personas pasen por el mismo calvario por el que pasamos mi abuela y yo.

Por tanto, podemos decir que EasyRent pretender evitar que otras personas sufran el actual proceso que hay que realizar para lograr alquilar un vehículo para personas con movilidad reducida.

Agradecimientos

En primer lugar, me gustaría agradecer a mi familia el apoyo por parte de mi familia desde el momento que decidí estudiar Ingeniería Multimedia.

También me gustaría agradecer a José Vicente Berná, que ha sido el que me ha guiado en este proyecto dando los consejos necesarios para llevar a cabo con éxito un proyecto de este estilo, aunque alguna vez no estuviéramos de acuerdo en algunas cosas.

Tampoco quiero dejar fuera a los profesores docentes que a lo largo de los 4 años de estudio ha impartido clase con más o menos acierto pero que en el fondo lo hacían por nosotros y que sin duda

Por último, pero sin duda no menos importante, quiero agradecer a mis amigos por aguantar mis quejas y los comentarios sobre la materia, que, aunque no entendierais nada siempre habéis estado ahí apoyándome y ayudándome tanto en los buenos como en los malos momentos.

Índice de contenidos

Licencia	2
Resumen	3
Motivación, justificación y objetivo general	5
Agradecimientos	7
Índice de figuras	12
Índice de tablas	14
1. Introducción	17
2. Estudio de viabilidad	20
2.1. Lean Canvan	20
3. Estado del arte.	24
3.1. Conceptos previos.	24
3.2. Problemática.	25
3.3. Acceso a ayudas técnicas.	26
3.4. Análisis de soluciones tecnológicas.....	29
3.4.1. AVIS.	29
3.4.2. Rentalcars.....	31
3.4.3. Rentaire.	35
3.5. Conclusiones.....	38
4. Objetivos.	40
5. Metodología.	42
6. Análisis y especificación	45
6.1. Usuarios de la plataforma.	45
6.2. Diagramas de casos de uso.	46
6.3. Requisitos funcionales.....	52
6.4. Requisitos no funcionales.	58
7. Diseño.....	61

7.1.	Diseño de la arquitectura conceptual	61
7.2.	Diseño de la arquitectura tecnológica.	62
7.3.	Diseño de los aspectos relativos al back-end.....	64
7.3.1.	Persistencia de datos.....	64
7.3.2.	Diseño API	68
7.3.3.	Aspectos de seguridad.	75
7.4.	Diseño de los aspectos relativos al front-end.	76
7.4.1.	Conceptos previos	76
7.4.1.1.	Principio de interfaz autodescriptiva.	76
7.4.1.2.	Principios del Flat design.....	77
7.4.2.	Diseño de interfaces.....	77
7.4.3.	Prototipos de EasyRent.	78
7.4.4.	Guías de estilo.	85
7.4.4.1.	Elementos generales.	86
7.4.4.2.	Paleta de colores.	86
8.	Implementación	89
8.1.	Entorno de desarrollo.	89
8.2.	Metodología empleada.	90
8.3.	Implementación del back-end.....	91
8.4.	Implementación del front-end.....	92
8.5.	Problemas durante el desarrollo.....	95
9.	Resultado final.....	98
9.1.	Home-Page.....	98
9.2.	Login/Registro.	100
9.3.	Perfil de usuario(empresa/Cliente).....	102
9.4.	Listado productos.....	104
9.5.	Detalle producto.	105
9.6.	Mis Reservas.....	107

9.7.	Detalle reserva.	109
10.	Conclusiones y trabajo futuro.	112
10.1.	Conclusiones.....	112
10.2.	Trabajo futuro.	113
	Referencias.....	115
	Apéndice I.....	117

Índice de figuras

Figura 1 : Lean Canvan para EasyRent	22
Figura 2 : Metabuscadores.....	25
Figura 3: Talonario de ejemplo.	27
Figura 4: Actividad del grupo.	28
Figura 5: Grupo de FaceBook: Donaciones de sillas de ruedas.....	28
Figura 6: Home-Page Avis.....	30
Figura 7: Resultado de una búsqueda en AVIS.....	30
Figura 8: Home-Page Rentalcars	32
Figura 9: Resultado de búsqueda Rentalcars.	33
Figura 10 Home-Page empresas Rentalcars.....	34
Figura 11: Home-Page Rentaire.	35
Figura 12: Resultado de busqueda en Rentaire.	36
Figura 13. Ejemplo del BackOffice de Magento.	38
Figura 14: Diagrama de casos de uso Cliente.....	47
Figura 15: Diagrama de casos de uso Empresa.	47
Figura 16: Diagrama casos de uso Administrador.....	48
Figura 17: Diseño conceptual EasyRent.	62
Figura 18: Stack tecnológico EasyRent.....	64
Figura 19: Modelo entidad relación de EasyRent.	68
Figura 20: Flujo de datos API Rest.....	69
Figura 21. Ejemplo de respuesta de la API.....	69
Figura 22: Bocetos EasyRent.	79
Figura 23: Wireframe global de EasyRent.....	80
Figura 24. Wireframe Home-page.....	81
Figura 25. Resultado de búsqueda y detalle de un producto.	82
Figura 26. Ver reservas.....	83
Figura 27. Wireframes para la gestión de productos.....	83
Figura 28. Páginas de perfil EasyRent.	84
Figura 29. Ver detalle empresa.	85
Figura 30. Menú responsive y tabla paginada.	86
Figura 31. Paleta de colores EasyRent.	87

Figura 32: Imagen corporativa. EasyRent	87
Figura 33: Logotipo de la plataforma	87
Figura 34. Sprint de EasyRent	90
Figura 35. Google-places.directive.ts.....	94
Figura 36. Ejemplo de input con la directiva.....	94
Figura 37. Resultado de la diectiva.	94
Figura 38 Home-page EasyRent.	99
Figura 39. Home-page móvil. Figura 40. Home-page móvil menú desplegado	100
Figura 41. Login EasyRent..	101
Figura 42. Registro EasyRent.....	102
Figura 43. Perfil empresa.	103
Figura 44. Detalle producto.....	106
Figura 45. Dialogo para añadir valoracion.	107
Figura 46. Mis reservas..	108
Figura 47. Detalle reserva.	109
Figura 48.Reserva en PDF.....	110
Figura 49. Ejemplo del filtro recibido.....	117
Figura 50. Método para generar la cláusula where.	117
Figura 51. Método general para obtener productos filtrados.	118

Índice de tablas

Tabla 2. Caso de uso 1.....	48
Tabla 3. Caso de uso 2.....	48
Tabla 4. Caso de uso 3.....	49
Tabla 5. Caso de uso 4.....	49
Tabla 6. Caso de uso 5.....	49
Tabla 7. Caso de uso 6.....	50
Tabla 8. Caso de uso 7.....	50
Tabla 9. Caso de uso 8.....	50
Tabla 10. Caso de uso 9.....	51
Tabla 11. Caso de uso 10.....	51
Tabla 12. Caso de uso 11.....	51
Tabla 13. Caso de uso 12.....	52
Tabla 14. Caso de uso 13.....	52
Tabla 15: Caso de uso 14.....	52
Tabla 16. Requisito funcional 1.....	53
Tabla 17. Requisito funcional 2.....	53
Tabla 18. Requisito funcional 3.....	53
Tabla 19. Requisito funcional 4.....	54
Tabla 20. Requisito funcional 5.....	54
Tabla 21. Requisito funcional 6.....	54
Tabla 22. Requisito funcional 7.....	55
Tabla 23. Requisito funcional 8.....	55
Tabla 24. Requisito funcional 9.....	55
Tabla 25. Requisito funcional 10.....	56
Tabla 26. Requisito funcional 11.....	56
Tabla 27: Requisito funciona 12.....	56
Tabla 28: Requisito funcional 13.....	57
Tabla 29: Requisito funcional 14.....	57
Tabla 30: Requisito funcional 15.....	57
Tabla 31: Requisito funcional 16.....	57
Tabla 32: Requisito funcional 18.....	58
Tabla 33. Requisito funcional 18.....	58

Tabla 34. Requisito no funcional 1	58
Tabla 35. Requisito no funcional 2	59
Tabla 36. Requisito no funcional 3	59
Tabla 37. Requisito no funcional 4	59
Tabla 38. Requisito no funcional 5	59

1. Introducción

Hoy en día es muy común el uso de aplicaciones y servicios web para buscar y comparar precios a la hora de comprar o alquilar cualquier producto, ya sea comprar un vuelo, una habitación de hotel para las vacaciones o incluso alquilar un vehículo para desplazarse. Este tipo de buscadores incorporan tecnologías de la información para poder realizar la comunicación empresa-cliente de una manera mucho más sencilla y cómoda para las dos partes.

Este tipo de buscadores se han hecho muy populares en poco tiempo debido principalmente al gran incremento que se ha producido en el uso de dispositivos con conexión a internet y a que ayudan al usuario a encontrar lo que están buscando de una manera mucho más cómoda y rápida que antaño, ya que, eliminan la necesidad de acudir a un establecimiento a consultar precios, horarios y demás información acerca de un producto que se desea alquilar o adquirir.

Además de poder consultar los precios e información acerca de los productos ofertados, estas plataformas incluyen métodos de pago seguros tipo PayPal que transmiten seguridad y confianza a los clientes. Esto implica una mayor reducción de la necesidad de acudir a los establecimientos, ya que, tareas cotidianas como son los cobros pasan a realizarse desde cualquier dispositivo gracias a internet, eliminando por ejemplo las tediosas colas de espera.

Cabe destacar que estas plataformas reducen el número de veces que acudimos a los establecimientos, pero resulta imposible eliminar por completo esta acción, debido a que, hay acciones que necesitan la presencia de los clientes (recoger/entregar el vehículo de alquiler...).

Una de las grandes virtudes de este tipo de buscadores es que no tenemos por qué ceñirnos a un horario convencional, ya que, este tipo de servicios están a nuestra disposición las 24 horas del día. Además, son accesibles desde cualquier dispositivo con conexión a internet, lo que permite hacer uso de ellos en cualquier lugar porque actualmente cualquier persona dispone de un *smartphone*, una *Tablet*, o de un ordenador portátil con conexión a internet.

Estamos hablando de buscadores como *booking*, *Avis*, *rentalcars*, *skyscanner*, etc. Estos buscadores además de ayudarnos a encontrar lo que deseamos nos facilitan la tarea de comparar precios de las diferentes empresas, ya que, con un resultado de una búsqueda sencilla obtenemos la información de diferentes anunciantes, así de un vistazo rápido poder contrastar la información de diferentes fuentes.

Normalmente, el uso de este tipo de buscadores no repercute demasiados gastos, ya que, anunciarse en este tipo de plataformas suele ser gratuito lo que permite a empresas pequeñas

y grandes darse a conocer de manera fácil y eficaz. Cabe destacar que en el caso de que estos sistemas incluyan cobros o cuotas por anunciarse puede repercutir en un mejor posicionamiento a la hora de mostrar los resultados a los clientes, o incorporar funcionalidades extras en la parte de administración que las cuentas gratuitas no disponen.

Por otro lado, no solo sirve con anunciarse, sino que estos buscadores premian el servicio recibido por los clientes, pongamos como ejemplo una pequeña empresa familiar que alquila estos vehículos y da un gran servicio a sus clientes que se lo agradecen realizando valoraciones positivas dentro del buscador propiciando que en búsquedas futuras los resultados de esta pequeña empresa aparezcan mejor posicionados que los demás.

Con todo lo anterior, este proyecto consiste en la creación de una herramienta software que permita a las empresas de alquiler de vehículos para personas con movilidad reducida anunciar sus productos, de manera que se encuentren englobados dentro de un buscador al que tendrá acceso cualquier tipo de usuario que desee alquilar este tipo de medio de transporte. De esta manera solventamos los problemas anteriormente mencionados a la hora de buscar una empresa que disponga de estos vehículos. Además, aumentamos las posibilidades de estos negocios, ya que mediante el uso de esta plataforma podrán aumentar el valor de su servicio y emplearlo como medio de difusión para captar así más clientes.

Esta herramienta software estará dividida en dos grandes bloques muy enfocados al usuario final que la empleará. Por un lado, una parte de administración donde las empresas darán de alta los productos que desean ofertar y ver sus reservas. Por otro lado, tendremos una parte pública donde los potenciales clientes podrán realizar las búsquedas a su gusto y visualizar la información.

2. Estudio de viabilidad

2.1. Lean Canvan

EasyRent es un proyecto que combina elementos de carácter tecnológico con una parte social y humana, ya que, pretende ayudar a esas personas que tengan la necesidad de hacer uso de alguna ayuda técnica para facilitar su movimiento.

Si contemplamos el ciclo de vida completo de este proyecto tendríamos que detallar tres fases o estados. En primer lugar, una fase de especificación en la que se realizarán todos los requerimientos que estableceremos más adelante. Por otro lado, una de desarrollo, despliegue y publicidad por el lanzamiento donde el objetivo será implementar la herramienta y darla a conocer para captar usuarios y por último la fase de mantenimiento y mejoras donde el objetivo principal será mantener satisfechos a los usuarios de la plataforma añadiendo nuevas funcionalidades demandadas por éstos.

Para analizar cada una de las fases descritas anteriormente, hemos empleado un diagrama Lean Canvan, el cual nos propone el siguiente análisis, para poder estudiar la viabilidad del producto.

En primer lugar, establece el **segmento de clientes**, es decir, cuáles son los clientes potenciales que harán uso de la plataforma y cuál es el nicho de mercado en el que se engloba la plataforma. Para EasyRent este segmento de clientes está formado por cualquier tipo de persona que necesite hacer uso de una ayuda técnicas o que esté relacionada con su comercialización, alquiler, préstamos o venta. Además, se requiere los usuarios de la plataforma tengan conocimiento del sector. Por tanto, podemos afirmar que en este caso que la edad no es un factor importante a la hora de hacer uso de la plataforma EasyRent.

Una vez analizado el segmento de clientes el Lean Canvan propone un análisis de los **Problemas** de estos clientes. Los principales problemas que tienen los clientes son en primer lugar la dificultad a la hora de encontrar los productos en la web, la poca difusión que tienen en la red y la necesidad que existe de alquilar los productos debido al elevado coste de comprarlos. Entraremos en más detalle en estos problemas en los siguientes apartados del proyecto.

El siguiente punto que establece el Lean Canvan es la **Propuesta de valor**, que define el valor del producto o servicio y que problemas resuelve. Para EasyRent esta propuesta de valor consiste en poner en contacto a las empresas de alquiler de ayudas técnicas con las personas que necesiten hacer uso de ellas.

El siguiente paso que propone Lean Canvan es aportar es comentar una **solución** a estos problemas. EasyRent será por tanto una herramienta que permita a las empresas comercializar sus productos a través de internet.

Una vez analizada la solución el siguiente paso en Lean Canvan es analizar cuáles son los **Canales**, es decir que medios vas a utilizar para comunicarte con tus clientes. En este caso el primer paso para establecer unos canales de comunicación es realizar una buena campaña de posicionamiento SEO para poder indexar tanto la herramienta como los contenidos que se van a mostrar dentro de ellas para que sea muy sencillo encontrarla a través de cualquier buscador. Una vez tengamos la plataforma bien posicionada se realizaran otro tipo de campañas para a más clientes y dar a conocer la herramienta para ellos se harán uso de campañas de tipo SEM[1], SMM[*Error! No se encuentra el origen de la referencia.*] y, Mailing[*Error! No se encuentra el origen de la referencia.*]....

A continuación, le toca el turno al apartado de **ingresos** y financiación de la empresa, en este caso EasyRent hará uso en primera instancia de un modelo gratuito subvencionado mediante la inclusión de publicidad dentro de la plataforma, con el objetivo de obtener ingresos en los inicios de la plataforma. Una vez EasyRent este asentado en internet y disponga de un tráfico de red considerable, este modelo gratuito pasara a una modalidad de pago por servicio obteniendo financiación a través de comisiones por cada transacción que se efectué dentro de la plataforma. El valor de esta comisión no se establecerá hasta que llegue el momento de implantarla y se harán cargo de ella los usuarios de tipo Empresa, es decir, los que van a comercializar a través de EasyRent.

Una vez analizado el apartado de ingresos Lean Canvan sugiere que analicemos el apartado de **Estructura de costes**, es decir que gastos va a suponer llevar a cabo la herramienta. Los principales gastos a la hora de llevar a cabo EasyRent serán las licencias de desarrollo y todo lo relacionado con el mantenimiento y la contratación del servidor y el dominio web y por supuesto la mano de obra que supondrá el desarrollo de la plataforma. En primera instancia este coste se puede calcular mediante una aproximación atendiendo a las horas de trabajo planificadas para desarrollar el proyecto. Es bueno realizar una aproximación del coste mediante las horas planificadas ya que, nos permite disponer de un presupuesto por si en un futuro fuera necesario comercializar el producto.

El siguiente paso que establece Lean Canvan es analizar **las métricas clave**, es decir, que aspectos de tu empresa debes medir y que aspectos o datos definen el modelo de negocio. Para

EasyRent las métricas más importantes serán el número de usuarios registrados, numero de productos ofertados, el feedback y opiniones que aportan los usuarios de la plataforma...

Por último, Lean Canvan sugiere realizar un análisis de la **ventaja competitiva**, que hace que EasyRent sea especial y diferente respecto a sus competidores, en este caso sería la única plataforma que ofrece un servicio especializado de cara al alquiler de ayudas técnicas.

A continuación, podemos observar el Lean Canvan completo para EasyRent.

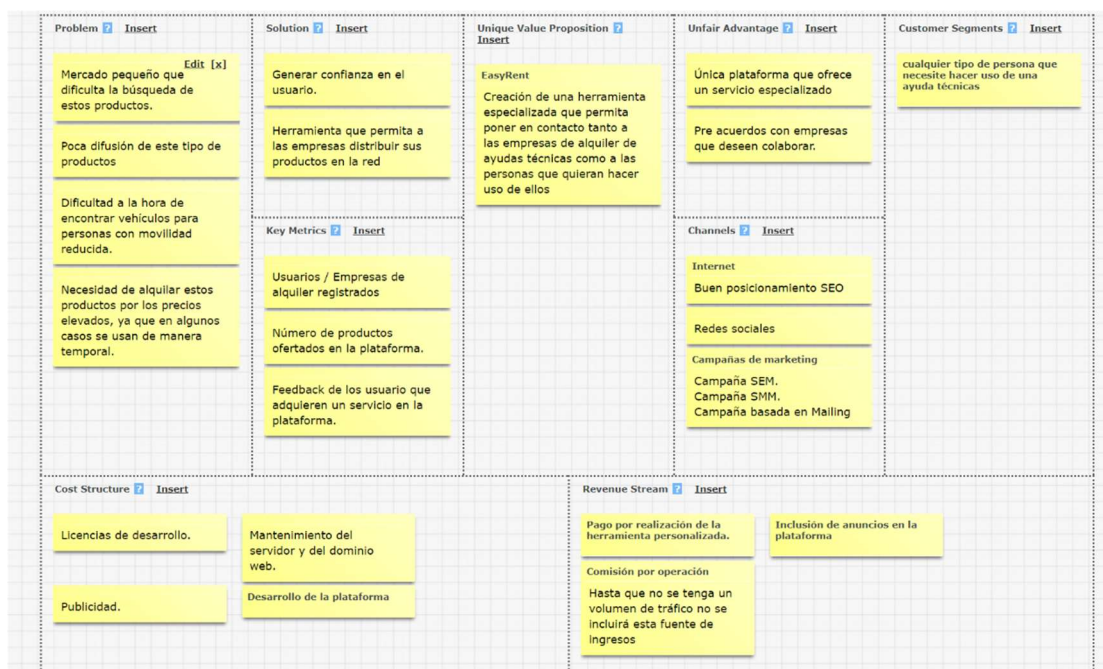


Figura 1 : Lean Canvan para EasyRent. <https://canvanizer.com/canvas/wKqIvViUpk6aP>

3. Estado del arte.

3.1. Conceptos previos.

Hoy en día la mayor parte de las personas busca cualquier tipo información en internet, ya sea un restaurante o una definición de una palabra, para ello hacen uso de motores de búsqueda o buscadores.

Actualmente existen multitud de motores de búsqueda en internet que nos permiten encontrar la información que deseamos, como mayor exponente de esta tecnología podemos destacar a Google, que recibe millones de consultas diarias en todo el mundo, siendo uno de los buscadores con mayor afluencia de usuarios, ya que permite obtener información de cualquier tipo.

Por tanto, un **motor de búsqueda o buscador** es un sistema informático capaz de obtener de manera ordenada información o archivos almacenados en servidores webs a partir de una consulta realizada por un usuario [4].

Estos buscadores siguiendo el ritmo de evolución de la tecnología han sufrido una mejora importante, ya que, han dado lugar a los famosos comparadores que realizan búsquedas obteniendo el mejor precio etc. Esta evolución de los motores de búsqueda es lo que conocemos como metabuscadores.

Un **metabuscador** es un sistema que es capaz de realizar consultas simultaneas en diferentes motores de búsqueda, analizando los datos resultantes y en función de un criterio propio muestran la información deseada, dando lugar a la aparición por ejemplo de los comparadores de precios mencionado anteriormente. Esto lo podemos observar en la Figura 2. [5]

A nivel tecnológico la principal diferencia entre estos dos tipos de buscadores es que normalmente un metabuscador no hace uso de bases de datos para almacenar la información que va a mostrar, debido a que, obtiene esta información de varios buscadores, es por lo que comúnmente un metabuscador es conocido como un buscador de buscadores, esto no implica que un metabuscador no haga uso de bases de datos propias para almacenar otro tipo de información.

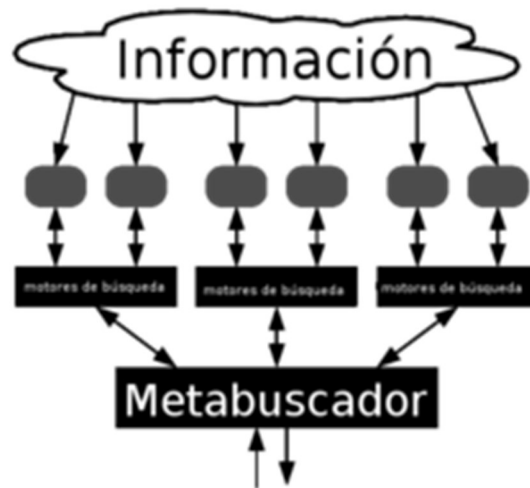


Figura 2 : Metabuscadores. <https://es.wikipedia.org/wiki/Metabuscador>

Por último, cabe destacar que la actual tendencia tanto para los buscadores como para los metabuscadores es hacer uso de la geolocalización para filtrar los resultados por proximidad, es decir, muestran la información que se encuentra más próxima al usuario que realiza la búsqueda, esto aporta una gran mejora en usabilidad, ya que, resulta mucho más sencillo obtener lo que se está buscando o ponerse en contacto con la persona o empresa que anuncia algo.

3.2. Problemática.

Actualmente el principal problema que encontramos para adquirir una ayuda técnica para personas con movilidad reducida es la falta de información sobre el acceso a estos productos y la escasa difusión en la red acerca de estas ayudas.

Hoy en día es muy usual recurrir a internet cuando nos surge una necesidad o un problema como este caso tener que hacer uso en un momento de tu vida de un andador o de una ayuda para poder caminar, el tipo de empresas que nos ofrecen estos servicios normalmente son pequeñas y a nivel local y no tienen la necesidad de invertir dinero para obtener presencia en internet, ya que se sustentan con la clientela tradicional y con los posibles acuerdos comerciales que puedan tener con hospitales y residencias para la tercera edad, es decir, son negocios que normalmente están asentados con su actual grueso de clientes.

Por otro lado, en el caso de que estas pequeñas empresas dispongan de contenido en la red normalmente no suele ser de mucha calidad y no suele estar bien indexado, lo que dificulta la búsqueda e implica inseguridad de los clientes hacia esa empresa.

Por otro lado, me gustaría destacar debido a la cercanía con municipios turísticos es que en ellos los problemas relacionado con el alquiler se acentúan en gran medida, debido a que estos negocios se centran en un cliente extranjero, esto implica por ejemplo que la información que empuen para la difusión este escrita en un idioma extranjero, dificultando así el proceso de compra o alquiler a una persona que desconozca el idioma. Además de esto, en los municipios turísticos este tipo de empresas suelen tener acuerdos comerciales con hoteles situando puntos de recogida/entrega en ellos para así atraer clientela fija gracias a los huéspedes.

También me gustaría destacar un problema al que se enfrenta personas con una renta baja a la hora de adquirir un producto de estas características, ya que, normalmente este tipo de productos suele tener un precio elevado lo que puede llegar a dificultar el acceso a las ayudas técnicas para personas con un poder adquisitivo bajo, sumado a la poca información acerca de las subvenciones o ayudas que ofrecen entidades públicas como pueden ser ayuntamientos, hospitales. Además de esto el proceso para acogerse a las ayudas a la dependencia suele ser largo y tedioso y normalmente tarda en hacerse efectiva esta ayuda.

3.3. Acceso a ayudas técnicas.

Como hemos mencionado anteriormente existe una dificultad a la hora de acceder a las ayudas técnicas como pueden ser sillas de ruedas, andadores o cualquier elemento que ayude a superar una discapacidad puntual o permanente.

Actualmente podemos acceder a ayudas técnicas de las siguientes maneras.

En primer lugar, podemos decidir alquilar un producto, ya que, por un lado, solo lo necesitamos durante un periodo de tiempo limitado o porque necesitemos probar el producto antes de adquirirlo, ya que, como hemos mencionado anteriormente suelen tener un precio elevado. Las empresas que dan este tipo de servicios a los clientes suelen ser pequeñas y basan su funcionamiento a partir de un stock de productos que se pueden reservar en función de la disponibilidad en el momento de realizar una reserva.

También me gustaría destacar por experiencia personal que estas empresas emplean métodos tradicionales de administración, ya que, para dejar constancia de un alquiler emplean los típicos talonarios con copia para hacer valida una reserva y la mayoría no se han planteado dar el salto tecnológico.

A continuación, podemos observar el tipo de recibo con el que dejan constancia del alquiler de una ayuda técnica.

El formulario es un recibo con el título 'RECIBO' en naranja. Incluye campos para el número del recibo (N.º), la fecha (de ___ de ___) y el nombre del receptor (RECIBI de ___). Hay un campo para la cantidad de Euros (la cantidad de Euros ___) y otro para el valor en Euros (Euros ___). Hay un espacio para la descripción (por ___) y un espacio para la firma y el sello (FIRMA Y SELLO:).

Figura 3: Talonario de ejemplo. Fuente: <https://www.amazon.es/Talonarios-Recibo-Praxton-Duplicado-Paquete/dp/B01ENMOC7C>

Por otro lado, podemos adquirir una ayuda técnica a través de centros especializados como puede ser una ortopedia donde si tenemos que hacer uso de estos productos por prescripción médica podemos obtener ayudas por parte del estado dependiendo del producto que adquiramos y de nuestro nivel de renta, ya que nos tendremos que acoger por ejemplo al plan de prestaciones específicas de la Comunidad Valenciana [6]. Estas ayudas se deben solicitar una vez comprado el producto por lo que implica un primer desembolso para hacernos con el producto.

Este tipo de empresas que se dedican a la distribución de productos especializados suelen basar su funcionamiento en un catálogo de una marca o varias marcas donde en la mayoría de los casos es necesario pedir el producto, ya que, no disponen stock de los productos más específicos.

Como dato importante que da muestra de que este tipo de empresas no han dado el salto a las nuevas tecnologías es la existencia de grupos o salas en redes sociales como *Facebook*, *Telegram*, *Twitter* donde la gente se pone en contacto para deshacerse o incluso donar por ejemplo de silla de ruedas o andadores que ya no necesitan por cualquier motivo.

A continuación, se aportan un par de enlaces a grupos de *Facebook* donde diferentes personas ofrecen sillas de ruedas:

- Venta de sillas de ruedas: www.facebook.com/groups/684711145238620/

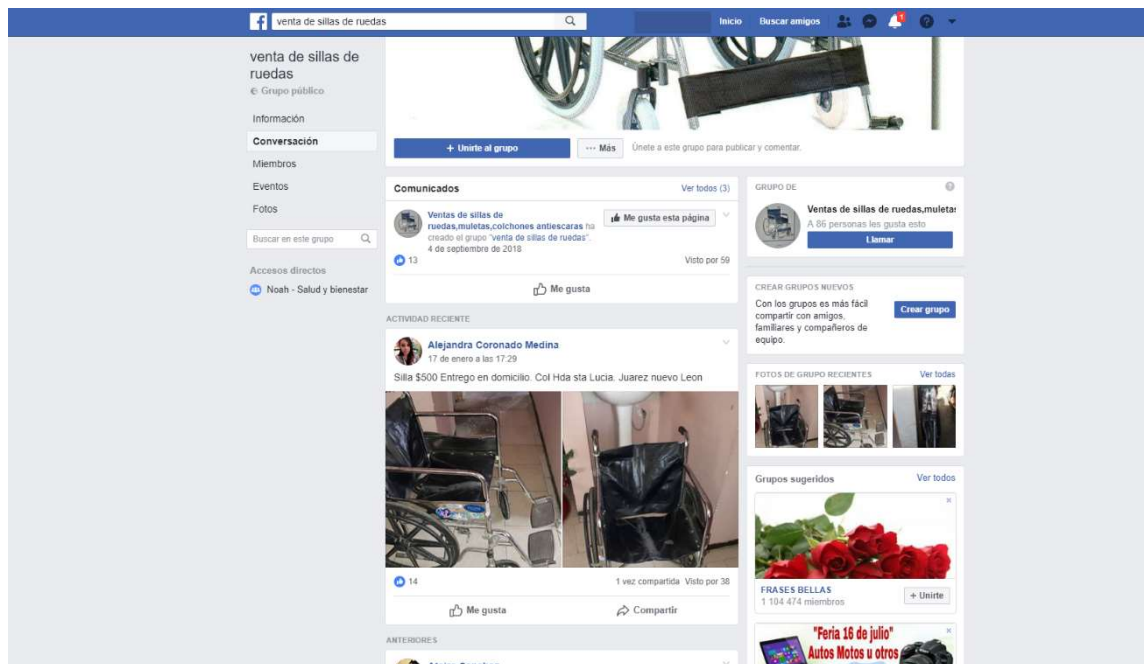


Figura 4: Actividad del grupo: Venta de sillas de ruedas de Facebook. Fuente: <https://www.facebook.com/groups/684711145238620/>

- Donaciones de sillas de ruedas: www.facebook.com/groups/421829764915696/

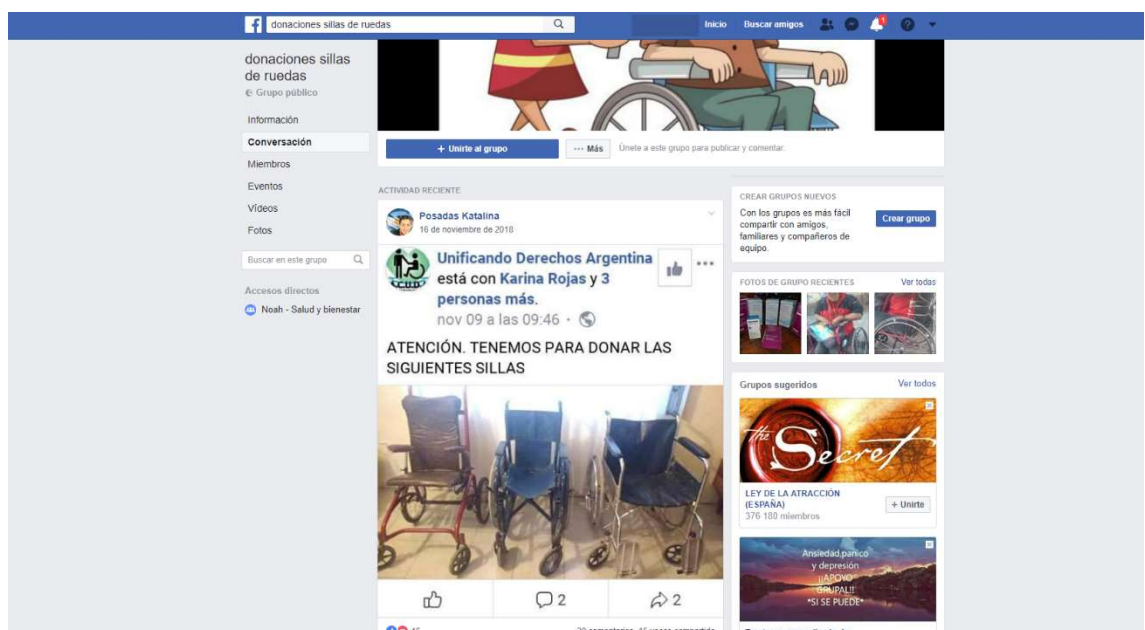


Figura 5: Grupo de Facebook: Donaciones de sillas de ruedas. Fuente: <https://www.facebook.com/groups/421829764915696/>

Por último, es necesario mencionar que también existe la posibilidad de adquirir este tipo de productos a través de las grandes plataformas de compra venta que existen en la actualidad como puede ser *milanuncios*, *Wallapop* o *eBay*. En estas plataformas se puede percibir el poco

mercado que existe actualmente, ya que, por ejemplo, no disponemos de una categorización clara ni una distinción entre las diferentes características que existen para estos productos (eléctrico, ruedas más grandes, tamaños especiales para viviendas pequeñas) por lo que para obtener resultados debemos buscar por la palabra exacta.

3.4. Análisis de soluciones tecnológicas.

A lo largo de esta apartado vamos a hacer un análisis de las soluciones tecnológicas que podemos encontrar actualmente a la hora alquilar un producto o servicio en la red. Cabe destacar que durante este análisis se van a seleccionar plataformas o empresas que no tienen por qué tratar este tipo de productos tan específico, pero que actualmente son muy conocidas y contemplan una necesidad similar a la problemática anteriormente comentada y por tanto pueden servir de inspiración o de modelo a la hora de resolver nuestro problema.

Actualmente existen diferentes soluciones tecnológicas a la hora de alquilar o adquirir un producto o servicio. En primer lugar, encontramos las tiendas directas donde una empresa expone únicamente sus productos o plataformas donde las empresas se dan de alta y pueden ofertar productos conjuntamente. Por otro lado, disponemos de soluciones tecnológicas mucho más complejas atendiendo a la definición de metabuscadores donde en este caso es la propia plataforma la encargada de recorrer la red en busca de información de otros sitios web para mostrarle al usuario final la información que más le interesa.

3.4.1. AVIS.

En primer lugar, me gustaría comentar la plataforma de alquiler de vehículos de la conocida marca AVIS. Esta plataforma la podemos clasificar atendiendo a lo descrito anteriormente como un buscador, en concreto como una tienda directa, ya que esta plataforma permite alquilar únicamente los vehículos ofrecidos por la marca propietaria de la plataforma, en este caso AVIS. A continuación, podemos observar una imagen de la *home-page* de AVIS.

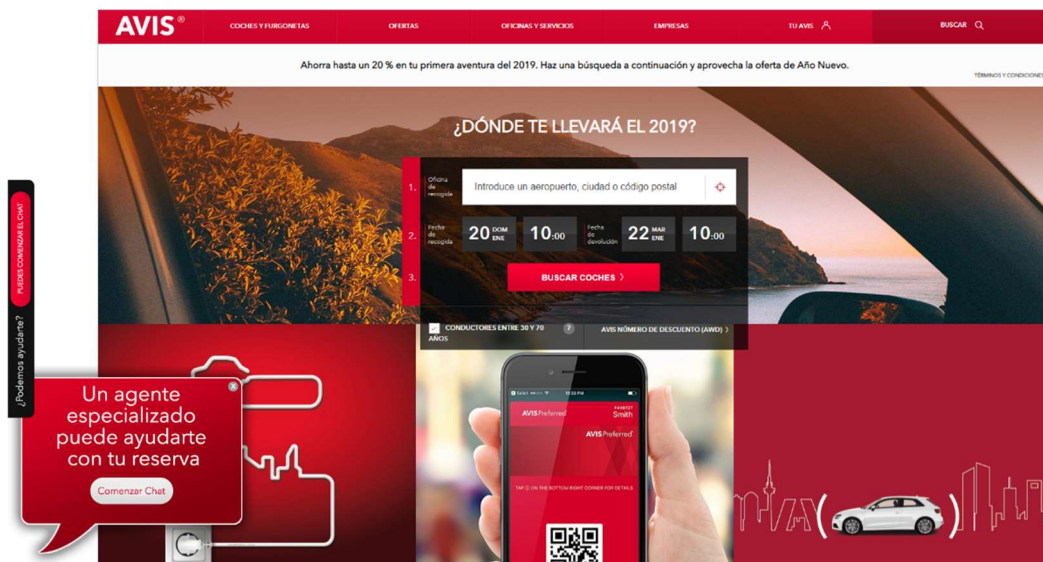


Figura 6: Home-Page Avis. Fuente: <https://www.avis.es/>

Esta plataforma ofrece un servicio de calidad a los clientes, ya que nos muestra una interfaz con un diseño muy cuidado y fácil de utilizar. El funcionamiento de esta plataforma gira en torno a un buscador central en el que realizamos nuestra consulta en función del lugar donde deseamos recoger el vehículo que vamos a alquilar. En la siguiente imagen podemos observar un resultado de una búsqueda, también podemos apreciar como el diseño de la interfaz también está muy cuidado en esta sección de la plataforma, ya que, por ejemplo, las imágenes de los vehículos son modelos muy fieles a la realidad y todos están en la misma disposición haciendo que la apariencia sea muy uniforme.

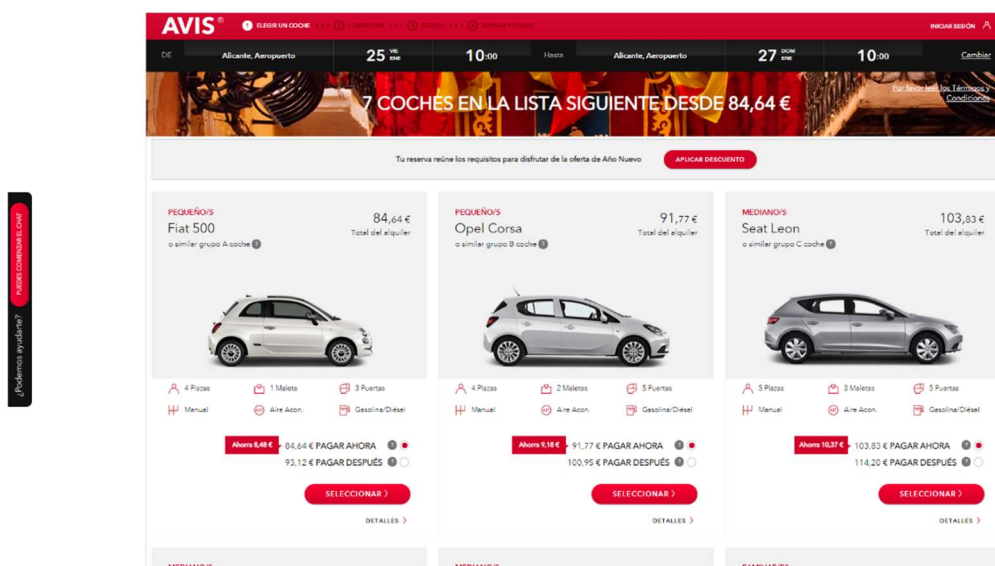


Figura 7: Resultado de una búsqueda en AVIS. Fuente: <https://www.avis.es/>

Como punto a favor de esta plataforma es la cantidad de información que muestra de los vehículos ofertados, ya que, el cliente tiene a su disposición la ficha del vehículo que está visualizando, este punto es muy importante porque facilita considerablemente tomar la decisión de que vehículo alquilar atendiendo a las necesidades específicas para cada cliente.

Por otro lado, de cara a la empresa el uso de este tipo de plataforma le permite obtener mayor difusión, ya que, está bien posicionada a nivel SEO y aparece en las primeras entradas en *Google* al realizar una búsqueda. Este incremento en la difusión permite llegar a muchas más personas lo que normalmente repercute en un aumento de los ingresos, ya que, es mucho más fácil alquilar un vehículo y por tanto tener más clientes. Por si esto fuera poco, esta empresa es una de las más importantes y con más presencia en la plataforma *Rentalcars* que analizaremos a continuación.

3.4.2. Rentalcars.

En segundo lugar, nos encontramos con la plataforma por excelencia para el alquiler de vehículos, *RentalCars*. Esta plataforma a primera vista puede parecer un metabuscador, ya que, oferta vehículos de multitud de marcas que disponen de su propia web de alquiler, pero entrando en más detalle sobre esta plataforma podemos observar que son las empresas las que deciden darse de alta en la plataforma para poder comercializar sus vehículos en una de las plataformas más grandes para el alquiler de vehículos.

Por otro lado, cabe destacar que este tipo de plataformas existe para mercados muy consolidados en internet y que mueven muchas transacciones a diario, como pueden ser la compra de vuelos, la reserva de hoteles o la contratación de seguros.

Este tipo de sistemas son realmente útiles para los clientes finales, ya que, con una simple búsqueda basada en unos criterios preestablecidos pueden obtener toda la información de la oferta actual de alquiler de vehículos facilitando así el proceso de comparación de precios y servicios de cada una de las compañías existentes.

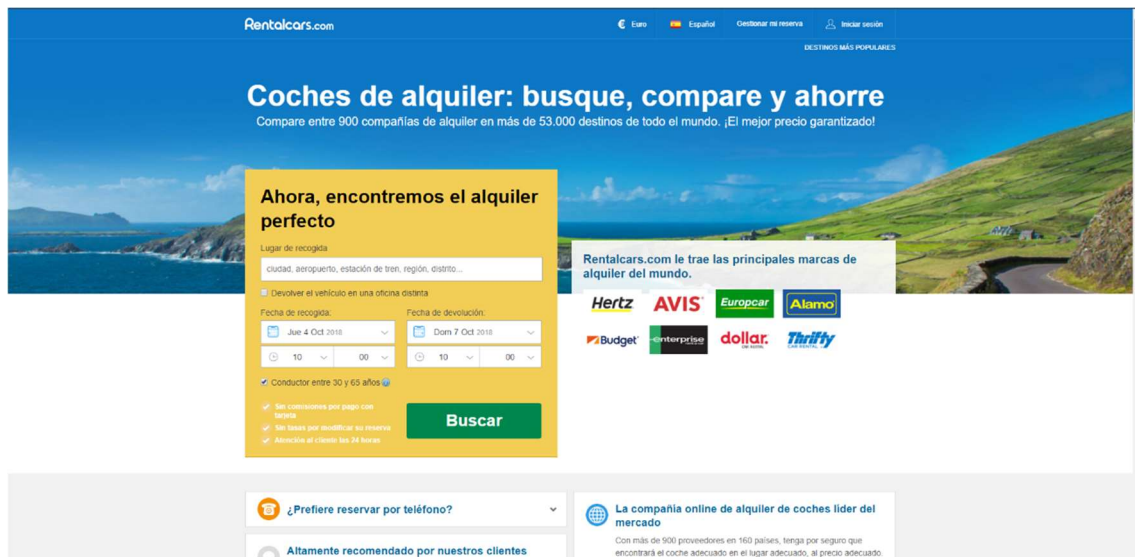


Figura 8: Home-Page Rentalcars, Fuente: <https://www.rentalcars.com/>

A continuación, podemos observar el resultado de una búsqueda realizada por un usuario que desea alquilar un vehículo donde podemos visualizar un listado de vehículos disponibles para alquilar de las diferentes empresas adheridas a esta plataforma.

De este resultado de búsqueda me gustaría destacar la cantidad de información y formas de filtrar que se le ofrecen al usuario, ya que esta plataforma permite dos tipos de búsqueda por un lado una más sencilla donde los usuarios únicamente deben realizar scroll hasta obtener el producto deseado y por otro lado ofrece la posibilidad de refinar mucho la búsqueda mediante una serie de filtros situados en la parte izquierda de la pantalla. Cabe destacar que esta solución es muy común entre plataformas de estas características. Como punto positivo me gustaría destacar que las dos formas de encontrar un resultado están muy bien integradas ya que la cantidad de filtros no interfiere en ningún momento una búsqueda normal.

Por último, en cuanto al resultado de búsqueda, me gustaría destacar la posibilidad de ver la valoración media de cada vehículo del listado, ya que, es un factor que puede marcar la diferencia a la hora de decidirse por el vehículo a alquilar porque una valoración alta puede generar confianza de cara al usuario condicionando en gran medida la elección final.

Ahora, encontremos el alquiler perfecto

Recogida

Modificar

Aeropuerto de Alicante (ALC)

Mié 13

Feb '19

10:00

Devolución

Aeropuerto de Alicante (ALC)

Sáb 16

Feb '19

10:00

Buscar

Ayuda

¿Desea realizar una consulta?

Popular

Compañía

Puntuación de la compañía

Excelente: 8+

84,95 €

Muy bueno: 7+

20,13 €

Ubicación de la compañía

En el aeropuerto

20,13 €

Autobús de cortesía

43,41 €

Política de combustible

Lleno/Lleno

22,79 €

Prepago (Reembolsable)

43,41 €

Prepago (reembolso parcial)

20,13 €

Características del coche

Asientos

Franquicia por daños

Depósito a bloquear en la oficina

Pagos con tarjeta durante la recogida

Kilometraje

Mostrar todos los resultados

La disponibilidad en Aeropuerto de Alicante (ALC) es menor durante estas fechas y los precios tienden a subir. Nuestro consejo: ¡no espere hasta el último minuto! Reserve ahora y aproveche una de nuestras ofertas.

Recogida

Aeropuerto de Alicante (ALC) - 10:00 13 de febrero de 2019

Devolución

Aeropuerto de Alicante (ALC) - 10:00 16 de febrero de 2019

Coches pequeños

Desde 20,13 €

Coches medianos

Desde 20,13 €

Coches grandes

Desde 21,78 €

Coches familiares

Desde 96,32 €

Coches de gama alta

Desde 49,16 €

Coches familiares

Desde 21,78 €

Coches SUV

Desde 49,16 €

Ordenar por:

Recomendado

Alquileres más recomendados primero

Precio

Alquileres más baratos primero

Ford Fiesta

o similar

Alta demanda: 44 reservados durante las últimas 24 horas

5 plazas

4 puertas

1 maleta grande

1 maleta pequeña

Aire acondicionado

Transmisión manual

Económico

Record

7.3

Muy bueno

(10000+ opiniones)

Política de combustible

Prepago (reembolso parcial)

Kilometraje ilimitado

Alicante - Aeropuerto

En el aeropuerto

Incluido en el precio:

Cambios en la reserva

Cobertura en caso de robo

Cobertura parcial por colisión (CDW)

Ver oferta

Guardar búsqueda

Añade Cobertura Total...

y protege tu alquiler

Renault Megane

o similar

Alta demanda: 26 reservados durante las últimas 24 horas

5 plazas

4 puertas

2 maletas grandes

1 maleta pequeña

Aire acondicionado

Transmisión manual

Compacto

Record

7.3

Muy bueno

(10000+ opiniones)

Política de combustible

Prepago (reembolso parcial)

Kilometraje ilimitado

Alicante - Aeropuerto

En el aeropuerto

Incluido en el precio:

Cambios en la reserva

Cobertura en caso de robo

Cobertura parcial por colisión (CDW)

Ver oferta

Guardar búsqueda

Opel Corsa

o similar

Alta demanda: 27 reservados durante las últimas 24 horas

4 plazas

4 puertas

1 maleta grande

1 maleta pequeña

Gran oferta

Precio por 3 días: 51,20 €

Figura 9: Resultado de búsqueda Rentalcars. Fuente: <https://www.rentalcars.com>

Este tipo de herramientas permite que cualquier empresa del sector pueda darse de alta de manera gratuita y pueda empezar a comercializar sus productos a través de internet, únicamente deberá pagar una pequeña comisión cada vez que se realice una transacción a través del metabuscador, en este caso un alquiler de un vehículo.

A continuación, podemos observar la página principal de la parte de *Rentalcars* destinada a las empresas que ofertan vehículos de alquiler en la plataforma:

33

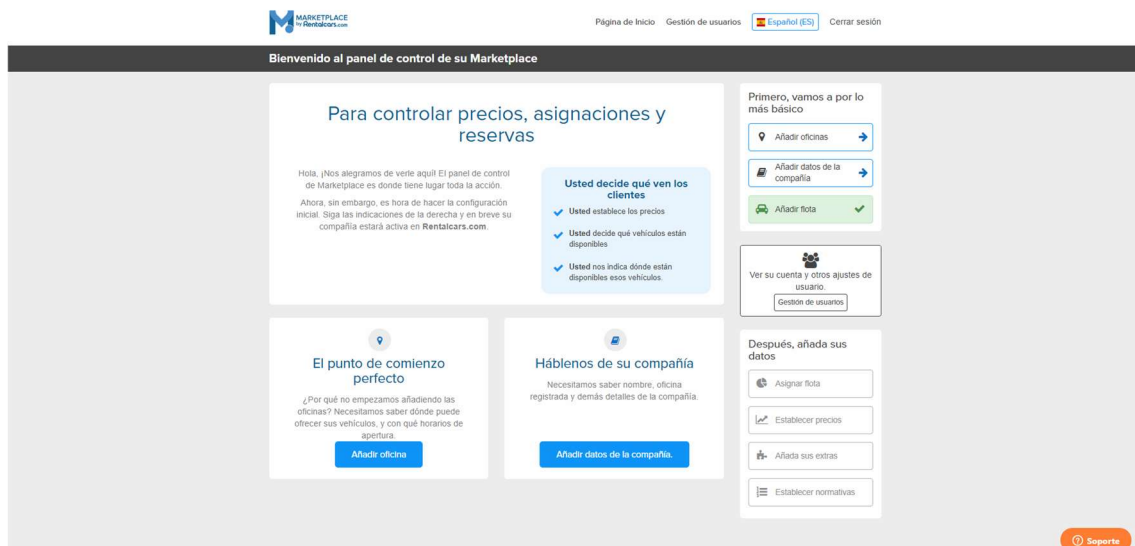


Figura 10 Home-Page empresas Rentalcars. Fuente: <https://marketplace.rentalcars.com/#/dashboard>

Esta parte de administración de *Rentalcars* permita a las empresas que decidan darse de alta introducir las oficinas de la que dispone, así como los vehículos de la flota que van a estar disponibles en cada una de las oficinas dadas de alta.

Como hemos mencionado anteriormente *Rentalcars* cobra comisión por cada transacción que se realice dentro de la plataforma, este porcentaje se puede visualizar en la parte de administración de la plataforma, ya que, en función de la demanda que haya en el lugar que se encuentra tu oficina este porcentaje será mayor.

Por otro lado, cabe destacar que esta herramienta de gestión permite ver la información de las reservas. Todo ello con una interfaz muy sencilla de utilizar y con un diseño limpio y atractivo.

El uso de este tipo de plataformas es de gran utilidad tanto para grandes empresas como para las más pequeñas, ya que, permite canalizar nuevos clientes aprovechando el buen posicionamiento SEO del que disponen los comparadores. También cabe destacar que este tipo de plataformas ofrecen a las compañías herramientas de gestión de los productos que van a ofertar dentro de la plataforma, así como la posibilidad de visualizar y gestionar sus reservas.

Las compañías que deciden adherirse a este tipo de plataformas normalmente ofertan solo una parte de sus productos para aumentar así las posibilidades de venta sin afectar a los métodos tradicionales de venta cuando por ejemplo un cliente acude a una oficina de alquiler de vehículos.

3.4.3. Rentaire.

Por último, me gustaría destacar el caso de *Rentaire* una pequeña empresa en comparación con las anteriores, *Rentaire* se dedica al alquiler y venta de maquinaria para la construcción. A nivel tecnológico podemos decir que es el caso más sencillo, ya que, está desarrollado mediante el uso de un CSM [7] orientado al e-commerce, es decir, se trata de una tienda online que ofrece la posibilidad de alquilar este tipo de maquinaria. En este caso al igual que la plataforma *Avis* es una tienda directa, ya que, solo podemos encontrar los productos que suministra una única empresa.

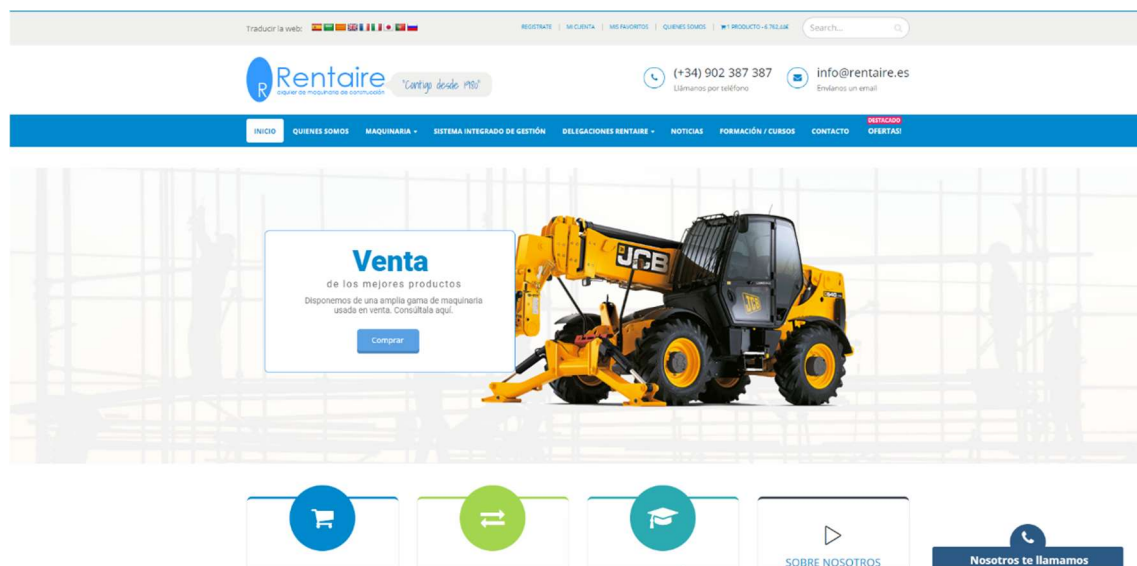


Figura 11: Home-Page Rentaire. Fuente: <https://www.rentaire.es/>

Como hemos mencionado en los casos anteriores, el uso de soluciones tecnológicas permite que esta empresa tenga una mayor difusión y en consecuencia un mayor número de clientes, incrementando así el volumen de ingresos

De cara al cliente de esta plataforma permite encontrar de una manera muy cómoda servicios y maquinaria pesada, a través de la red, ya que, presenta la información muy bien categorizada y de fácil acceso, además de contar con un buscador que permite localizar productos más concretos.

Si realizamos una búsqueda en esta plataforma lo primero que nos llama poderosamente la atención es la imposibilidad de buscar directamente mediante un rango de fechas para obtener el precio total del alquiler y por otro lado la gran diferencia en cuanto a diseño respecto a las dos plataformas anteriores a la hora de mostrar los resultados, ya que en este caso el diseño no es tan limpio como los anteriores.

Los resultados de una búsqueda realizada en esta plataforma únicamente nos muestran el título del producto, el precio de manera un tanto ambigua, ya que no especifica como se aplica (por días, semanas ...) y una valoración media a partir de unas estrellas.

A continuación podemos observar y corroborar lo anterior con este ejemplo de un resultado de una búsqueda realizado en la plataforma *Rentaire*:

The screenshot shows the Rentaire website interface. At the top, there's a navigation bar with links like 'INICIO', 'QUIENES SOMOS', 'MAQUINARIA', etc. Below this is a search bar and a sidebar with filters for 'MAQUINARIA DISPONIBLE'. The main content area displays a grid of product listings for hand tools. Each listing includes a product image, a title, a price, a rating (stars), and a 'Leer más' button. The products shown include various types of drills, impact wrenches, and saws. The prices are listed in Euros (€) and include VAT (IVA). The interface also features a 'FILTRO POR PRECIO' slider and a 'PRODUCTOS VISTOS RECIENTEMENTE' section at the bottom.

Figura 12: Resultado de búsqueda en Rentaire. Fuente: <https://www.rentaire.es/>

Este tipo de solución tecnológica es completamente válida para cualquier tipo de sector, ya que permite comercializar productos muy concretos en la red, además de esto, la ventaja que ofrecen los CSM es la parte de gestión, ya que nos permite administrar toda la plataforma con

un completo backoffice[8], donde el administrador o propietario de la plataforma podrá organizar de manera fácil y sencilla los productos que oferta así como introducir nuevos, visualizar en este caso las reservas que tiene activas. Además de esto, los CMS ofrecen información monetaria de la empresa (ingresos, gastos...), pero la funcionalidad no queda ahí, ya que, presentan multitud de extensiones o plugins para adecuar o añadir nuevas funcionalidades más específicas en función de cada caso.

En la Figura 13. Ejemplo del BackOffice de Magento. Fuente: <https://www.magento.com> podemos observar un BackOffice de ejemplo de uno de los CMS para el e-commerce más populares del momento, Magento, que podría ser perfectamente el mismo que emplea la plataforma de alquiler y venta *Rentaire*.

Este back-office es una potente herramienta web que permite la gestión completa de una tienda online, desde aspectos como la administración de los productos hasta configuraciones de los diferentes métodos de pago disponibles en la web.

En cuanto a la administración de los productos esta herramienta nos ofrece muchas posibilidades como la categorización libre de los productos, realizar promociones sobre estos, gestionar el stock disponible, crear productos personalizables estableciendo atributos propios como la talla o el color o incluso realizar volcados masivos.

Por otro lado, esta herramienta dispone posibilidades de venta internacional, ya que por ejemplo podemos configurar diferentes tipos de impuestos, envíos, etc.

También de este back-office me gustaría destacar que ofrece la posibilidad de crear nuevas páginas web o editar el contenido de ellas permitiendo una personalización completa añadiendo por ejemplo banners con ofertas por un tiempo limitado dando la posibilidad al dueño de la tienda personalizar la web sin necesidad de tener conocimientos de programación.

En definitiva, se trata de una potente herramienta de gestión y manipulación de tiendas online, cuyo funcionamiento y puesta en marcha debe ser guiada por profesionales del sector que conozcan la herramienta para sacar el máximo provecho de la infinidad de posibilidades que ofrecen este tipo de soluciones tecnológicas.

Por si todo esto fuera poco el back-office de *Magento* como cualquier CSM que se precie tiene la posibilidad de añadir más funcionalidades mediante el uso de plugins y temas que permiten realizar configuraciones más específicas.

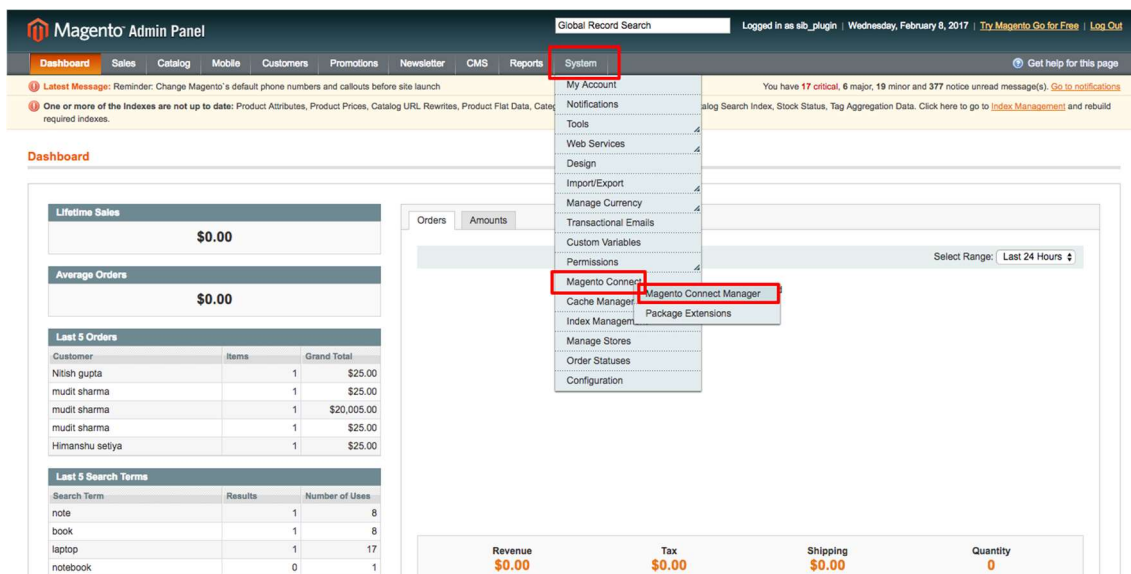


Figura 13. Ejemplo del BackOffice de Magento. Fuente: <https://www.magento.com>

Podemos afirmar por tanto que esta solución sería en primera instancia el primer paso para una transformación digital de una empresa.

3.5. Conclusiones.

Como hemos comentado anteriormente el uso de soluciones tecnológicas, en la mayor parte de los casos implica beneficios tanto para el cliente como para la empresa que decide subirse al carro de la digitalización, ya que, con una inversión en tecnología ven incrementados sus beneficios de manera notable gracias a la difusión que se puede hacer hoy en día a través de internet.

Actualmente existen multitud de soluciones tecnológicas válidas para cualquier tipo de empresa que decida comercializar sus productos a través de internet. La elección entre una y otra dependerá en primer lugar del tipo de mercado en el que se encuentren los productos que se desea comercializar a través de la red y en segundo lugar del nivel de digitalización que las empresas están dispuestas asumir, así como el desembolso económico que puedan permitirse.

De las plataformas anteriormente analizadas me gustaría destacar que todas tienen en común un buscador sencillo y fácil de utilizar con un diseño bastante cuidado y atractivo, ya que, hoy en día este tipo de detalles pueden marcar la diferencia entre el fracaso y el éxito en internet.

4. Objetivos.

El objetivo principal de este proyecto es proporcionar unos mecanismos de comunicación que permita poner en contacto a personas que necesiten hacer uso ayudas técnicas (andadores, scooter adaptadas...) con empresas que alquilan este tipo de productos, que no requiera grandes conocimientos en informática para poder ser usada. Este sistema en un primer lugar está destinado a PYMES que deseen ampliar su volumen de ventas de una manera fácil, sencilla y sin costes. En otras palabras, acercar la oferta de estas ayudas técnicas a la actual demanda que existe de ellos. Para ello debemos realizar un análisis de todos los requerimientos para poder definir esta herramienta de comunicación y gestión.

A continuación, se especifican una serie de subobjetivos más concretos en el devenir de este proyecto:

- Ayudar a difundir y dar a conocer los productos que oferten empresas de alquiler de ayudas técnicas.
- Analizar y comprender el mercado actual de este tipo de productos para conocer sus características y poder aportar una solución más acertada.
- Realizar un estudio de soluciones actuales que tratan problemas y situaciones parecidas a la de este proyecto para seguir una estrategia de éxito para resolver las necesidades que presenta este proyecto.
- Realizar un análisis del tipo de usuario atendiendo a la capacidad tecnológica y los conocimientos que tienen de ella para poder adecuar la solución al usuario al que va destinado.
- Crear una herramienta que permita a las empresas de alquiler de ayudas técnicas dar a conocer sus productos entre los diferentes usuarios y dotarles de un sistema de gestión de sus reservas.
- Analizar las diferentes posibilidades de monetización que tenemos disponibles y que más se adecuen a este proyecto.

5. Metodología.

A la hora de afrontar un proyecto de estas características es conveniente conocer las metodologías de desarrollo software que existen en la actualidad y como hemos pasado de unas metodologías tradicionales a unas ágiles que facilitan el proceso de desarrollo de proyectos software.

Al inicio del desarrollo software se empleaban **metodologías tradicionales** con un ciclo de vida en cascada, es decir, un proyecto se divide en fases y hasta que no se termina una de ellas no se puede empezar con la siguiente, además de esto en el momento que una fase ha finalizado no se puede volver hacia atrás. El principal problema de este tipo de metodología poca capacidad de adaptación a posibles cambios.

La tendencia actualmente es dejar a un lado las metodologías tradicionales para hacer uso de **metodologías ágiles** que están basadas en un ciclo de vida iterativo e incremental, es decir, se entregan prototipos con novedades visibles para el usuario final añadiendo funcionalidades nuevas en cada iteración, de esta manera es más sencillo recibir feedback por parte de los usuarios y poder así adaptarse a nuevas necesidades que puedan ir surgiendo durante el desarrollo de un proyecto. Estas iteraciones o sprints normalmente suelen tener poca duración (15 – 20 días) [9].

Las metodologías ágiles más empleadas actualmente son:

- SCRUM: está basado en sprints de duración de 2 a 4 semanas donde se establecen una serie de tareas que se deben realizar durante el sprint y al final de éste debe de tenerse una versión en producción visible al usuario final. Una vez finalizado el sprint se realiza una reunión de retrospectiva para evaluar el trabajo realizado y una última reunión para planificar el siguiente sprint.
- Kanban: está basado en la clasificación las tareas en función del estado en el que se encuentran, en un inicialmente todas empiezan en “por hacer”, conforme se van completando las tareas pasan a “resueltas”. Normalmente estas tareas las eligen los miembros del equipo de trabajo por turnos.
- Extreme Programming (XP): esta metodología fomenta a ser capaz de adaptarse a los cambios de requisitos en cualquier punto de vida del proyecto, es decir realizando análisis y especificación de requisitos durante el día a día, introduciendo al cliente con el equipo de desarrollo, por otro lado, esta metodología recomienda la programación de se realice por parejas, para generar código de más calidad, ya que, se evalúa y se

discute mientras se escribe. Al igual que las anteriores esta metodología se basa en el desarrollo iterativo e incremental realizando pequeñas mejoras o actualizaciones visibles al cabo de un periodo de tiempo planificado [10].

A la hora de decantarse por una metodología tradicional o ágil no podemos afirmar cuál de ellas es mejor que otra, ya que, están muy condicionadas al tipo de proyecto y a la manera de trabajar de cada uno. En el caso de este proyecto podemos decir que hacemos uso de dos tipos de metodologías, ya que, al principio del proyecto durante la especificación de este hacemos uso de una metodología más tradicional en cascada, es decir hasta que no terminamos un apartado a sección no iniciamos el siguiente.

Por otro lado, para la fase de desarrollo e implementación de la plataforma hare uso de una metodología ágil SCRUM realizando sprint de 2-3 semanas con una serie de tareas que se deberán realizar y ser visibles en una versión de preproducción. He escogido este tipo de metodología porque me parece la más adecuada para la fase de desarrollo debido a que te autoexige a completar una serie de tareas en un tiempo determinado, además de esto, como la filosofía SCRUM exige que al final de cada sprint exista una versión que pueda ser entregada a un cliente que en ese caso será mi tutor del proyecto, será mucho más fácil tener un feedback de cómo va el desarrollo y de posibles mejoras.

Como este proyecto se realiza de forma individual no es especialmente compleja la gestión del proyecto aun así voy a hacer uso de unas herramientas gratuitas que me faciliten el proceso. En primer lugar, hare uso de Toggl [11] para contabilizar las horas trabajadas para cada tarea, para poder tener de manera global un análisis de tiempo dedicado a cada tarea y al proyecto.

Por otro lado, para la planificación de los sprints anteriormente mencionados empleare la conocida herramienta Trello para clasificar las tareas de cada sprint en tres fases *TODO*, *DOING*, *DONE* (por hacer, haciendo, hecho). De esta manera puedo visualizar de manera rápida las tareas de un sprint y ver su estado.

Por último, me gustaría destacar que he elegido estas herramientas porque son gratuitas y el volumen de trabajo de este proyecto no va a exigir el uso de herramientas más profesionales como pueden ser Jira [12] o Microsoft Project [13].

6. Análisis y especificación

A lo largo de esta sección se definirán las funcionalidades y los requisitos del proyecto. Para ello se ha decidido emplear el estándar IEEE 830, ya que, este estándar nos permitirá seguir un modelo para definir un proyecto que nos asegura que se abordaran todas las áreas del proyecto, desde los usuarios que harán uso de la plataforma a las funcionalidades del sistema.

Este análisis de requerimientos consiste en una descripción completa del sistema que se va a desarrollar, con todas sus interacciones y funcionalidades, para ello siguiendo con el estándar IEEE 830 definiremos una serie de requisitos:

- Por un lado, los requisitos funcionales que describen el conjunto de entradas, comportamientos y salidas que el sistema va a proporcionar. Cada uno de ellos cuenta con una descripción de los pasos a seguir para llevar a cabo la funcionalidad. [14]
- Por otro lado, los requisitos no funcionales que definen las características que hacen referencia al comportamiento del sistema. Son empleados para imponer las diferentes restricciones, tanto en diseño como en funcionalidad. [15]

6.1. Usuarios de la plataforma.

Para poder especificar todos los requisitos funcionales y no funcionales es necesario conocer los diferentes tipos de usuarios y funcionalidades principales de cada uno de ellos dentro de la plataforma, para poder asignar los requisitos a cada tipo de actor involucrado.

Antes de realizar la clasificación de los usuarios que van a hacer uso de la plataforma podemos establecer dos estados que van a limitar las funcionalidades de cada uno de ellos, por un lado, el estado **identificado**, es decir, cuando el usuario ha hecho login en la plataforma introduciendo su usuario y contraseña. Por otro lado, el **no identificado** que todavía no ha hecho login o no se ha registrado, pero ha accedió a la web para buscar información.

En primer lugar, disponemos de los usuarios de tipo **Cliente**, los cuales están interesados en consumir los productos ofertados en la plataforma, estos usuarios podrán buscar un producto, visualizar su información y sus opiniones además de la información de la empresa que lo suministra, pero si decide alquilar el producto deberá encontrarse en el estado de **identificado** para poder realizar las funcionalidades más avanzadas del sistema como puede ser la reserva

del producto o dejar una valoración. De esta manera la empresa que suministra los productos obtiene la información sobre el usuario que quiere hacer uso de sus ayudas técnicas. Además de esto añadimos seguridad antibots, ya que, al ser necesario hacer login en la plataforma para realizar las reservas los bots tendrán un primer impedimento a la hora de realizar reservas de manera masiva.

Por otro lado, la plataforma dispondrá de un usuario de tipo **Empresa**, que es el encargado de publicar contenido en la plataforma, en este caso los productos que desea comercializar a través de ella. Por tanto, este tipo de usuario será el encargado de dar alta los productos que quiera anunciar a través de la plataforma, así como ver las reservas de cada uno de ellos. El usuario de tipo empresa también podrá visualizar los comentarios y/o valoraciones realizadas por los clientes con el objetivo de recibir un feedback de los usuarios que hagan uso de sus ayudas técnicas. Además de esto, el usuario tipo Empresa podrá registrar reservas, lo que le permitirá hacer uso de la plataforma desde el puesto de trabajo en la oficina de alquiler dando la posibilidad al usuario que usa la herramienta como medio de gestión de reservas.

Cabe destacar que este tipo de usuario deberá estar identificado para realizar las funcionalidades anteriormente descritas.

Por último, la plataforma dispondrá de un tipo de usuario **Administrador** que será el responsable de la plataforma, es decir, podrá ver los usuarios y las empresas registradas y demás información respectiva a la gestión de la plataforma. Este usuario será el encargado de revisar el correcto funcionamiento de herramienta por parte de los dos tipos de usuarios anteriormente descritos.

6.2. Diagramas de casos de uso.

Para comprender mejor las funcionalidades anteriormente descritas para los usuarios que va a disponer la plataforma se han realizado una serie de diagramas de casos de uso para cada tipo de usuario, con el fin de entender mejor su funcionamiento.

A continuación, podemos observar los diferentes diagramas de casos de uso, en primer lugar, el diagrama de caso de uso para un usuario de tipo **Cliente**:

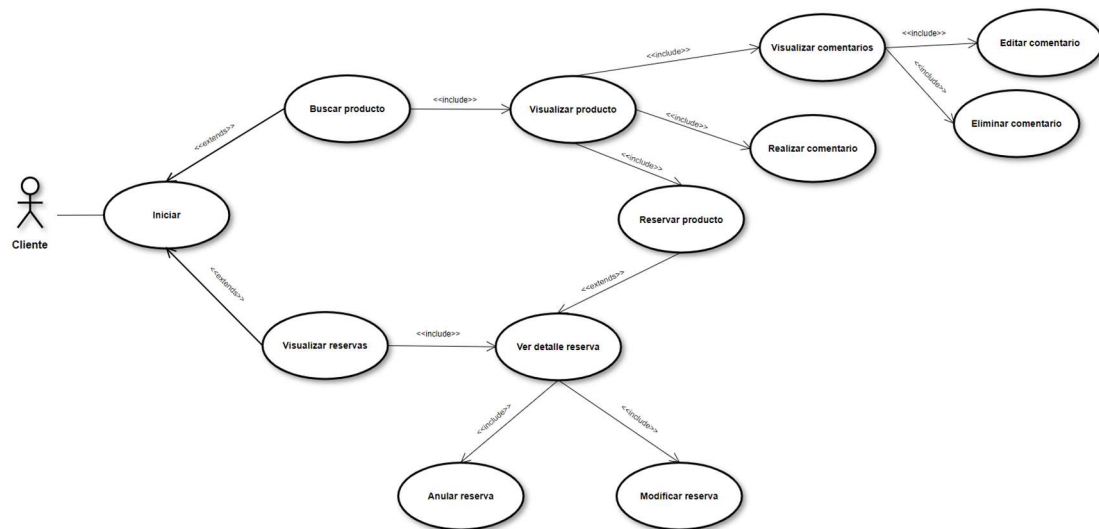


Figura 14: Diagrama de casos de uso Cliente. Fuente propia

Por otro lado, tenemos el diagrama de caso de uso para un usuario de tipo **Empresa**:

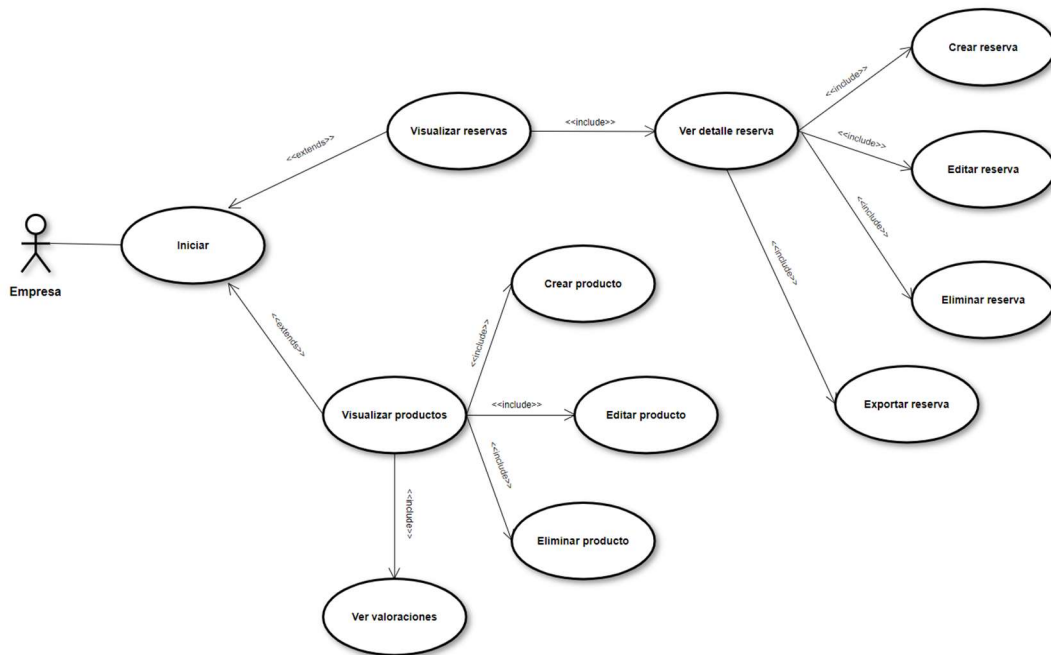


Figura 15: Diagrama de casos de uso Empresa. Fuente propia

Por último, podemos observar el diagrama de caso de uso para un usuario de tipo **Administrador**:

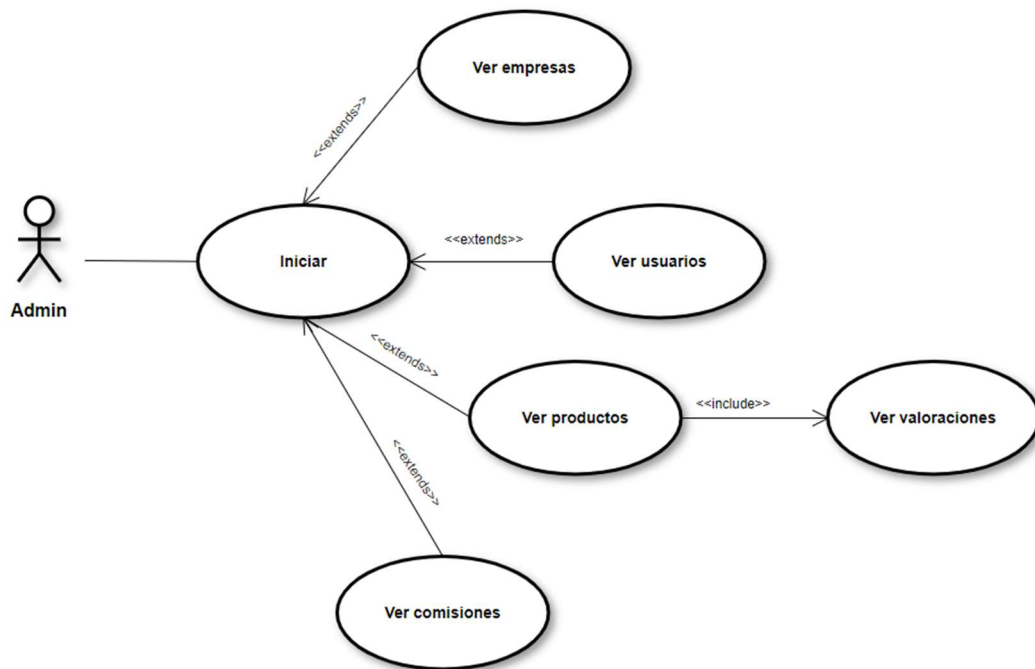


Figura 16: Diagrama casos de uso Administrador. Fuente propia

Identificador	CU-01
Nombre	Iniciar
Precondición	El usuario abre la aplicación con conexión a internet.
Secuencia	La aplicación se abre.
Postcondición	Al abrirse, se muestra la home-page de EasyRent.

Tabla 1. Caso de uso 1.

Identificador	CU-02
Nombre	Buscar producto
Precondición	El usuario introduce los criterios de búsqueda que desee como puede ser el nombre, el tipo de ayuda técnica o la empresa a la que pertenece.
Secuencia	<ul style="list-style-type: none"> • El usuario inserta los criterios de búsqueda. • Se realizan las peticiones a la API • Se devuelven los resultados.
Postcondición	Si hay resultados, se muestrean con el formato correcto.

Tabla 2. Caso de uso 2

Identificador	CU-03
Nombre	Visualizar producto
Precondición	El usuario realiza una búsqueda en el sistema y selecciona uno de ellos para ver sus detalles
Secuencia	Se muestran los detalles de un producto como puede ser una descripción, un precio por alquilar, la empresa a la que pertenece, etc. Así como una opción para reservarlo y para ver sus opiniones
Postcondición	-

Tabla 3. Caso de uso 3

Identificador	CU-04
Nombre	Reservar producto
Precondición	El usuario está visualizando el detalle de un producto y ha seleccionado un rango de fechas para realizar su reserva.
Secuencia	<ul style="list-style-type: none"> • El usuario introduce las fechas deseadas para su reserva • Se realizan las peticiones a la API para comprobar que está disponible y se confirma la reserva.
Postcondición	El usuario podrá visualizar la información de la reserva

Tabla 4. Caso de uso 4

Identificador	CU-05
Nombre	Realizar comentario
Precondición	El usuario identificado está visualizando un producto concreto y rellena el formulario para dejar constancia de un comentario/valoración
Secuencia	El usuario rellena los datos y se realiza una petición para almacenar la información
Postcondición	Una vez se ha guardado la información este comentario aparecerá en la sección de comentarios con las opciones para eliminar y editar.

Tabla 5. Caso de uso 5

Identificador	CU-06
Nombre	Visualizar reservas (Cliente)
Precondición	El usuario de tipo cliente accede a la sección de sus reservas.
Secuencia	La aplicación muestra un listado con todas las reservas del usuario identificado en la plataforma.
Postcondición	El sistema permite pulsar sobre una de ella para mostrar los detalles, así como las opciones de anular o modificar la reserva.

Tabla 6. Caso de uso 6

Identificador	CU-07
Nombre	Visualizar reservas (Empresa)
Precondición	Un usuario de tipo empresa accede a la sección de reservas
Secuencia	La herramienta muestra un listado con todas las reservas de la empresa que ha iniciado sesión, así como unos filtros.
Postcondición	La aplicación permite la opción de aplicar filtros para encontrar más fácilmente la información, así como la posibilidad de ver más detalles de la reserva, editarla o eliminarla.

Tabla 7. Caso de uso 7

Identificador	CU-08
Nombre	Crear reserva
Precondición	Un usuario de tipo empresa decide crear una reserva
Secuencia	Se rellenan los datos necesarios para una reserva, para si poder atender a un cliente que llegue al establecimiento.
Postcondición	La reserva se hace valida y podrá visualizarse en el apartado de reservas de la plataforma.

Tabla 8. Caso de uso 8

Identificador	CU-09
Nombre	Exportar reservas
Precondición	Un usuario de tipo empresa decide exportar una o varias reservas
Secuencia	El usuario puede seleccionar únicamente una reserva para exportarla o seleccionar un listado de reservas que desea exportar a PDF
Postcondición	La plataforma genera un PDF donde podrá visualizar la información de las reservas.

Tabla 9. Caso de uso 9

Identificador	CU-10
Nombre	Visualizar productos
Precondición	Un usuario de tipo empresa o administrador accede a la sección de productos
Secuencia	La aplicación muestra un listado con todos los productos dados de alta en el sistema para la empresa que hace el login. En el caso de que acceda un usuario de tipo administrador visualizará un listado con todos los productos dados de alta en la plataforma. Así como una serie de filtros para obtener información más concreta.
Postcondición	El sistema permite acceder a un producto para visualizar la su información, así como acciones más específicas como editar, eliminar o ver valoraciones.

Tabla 10. Caso de uso 10

Identificador	CU-11
Nombre	Crear producto
Precondición	Un usuario de tipo empresa pulsa en crear producto
Secuencia	La aplicación muestra un formulario donde el usuario deberá introducir la información que desee.
Postcondición	Una vez se crea el producto correctamente se podrá visualizar en el listado de productos de una empresa y será ofertado en la plataforma y los demás usuarios podrán ver su información al realizar una búsqueda.

Tabla 11. Caso de uso 11

Identificador	CU-12
Nombre	Ver empresas
Precondición	Un usuario de tipo administrador accede a la sección de ver empresas
Secuencia	La aplicación muestra un listado con todas las empresas dadas de alta en la plataforma.
Postcondición	El usuario podrá hacer clic en cada una de ellas para visualizar más información.

Tabla 12. Caso de uso 12

Identificador	CU-13
Nombre	Ver usuarios
Precondición	El usuario de tipo administrador accede a la sección de ver usuarios.
Secuencia	La aplicación muestra un listado con todos los usuarios registrados de la plataforma, así como un filtro para poder obtener información más concreta.
Postcondición	El usuario podrá hacer clic en cada uno de ellos para visualizar más información.

Tabla 13. Caso de uso 13

Identificador	CU-14
Nombre	Ver comisiones
Precondición	El usuario de tipo administrador accede a la sección de ver comisiones.
Secuencia	La aplicación muestra un listado con las comisiones generadas por cada una de las reservas que se realizan dentro de la plataforma.
Postcondición	El administrador podrá hacer clic en cada una de ellas para visualizar la información de la reserva.

Tabla 14: Caso de uso 14

6.3. Requisitos funcionales.

Antes de detallar todos los requisitos funcionales cabe destacar que cada uno de ellos tendrá un identificador único, un tipo de usuario que hemos comentado anteriormente, un nombre con una descripción para entender perfectamente a que hace referencia, los requisitos lógicos y un tipo de prioridad.

Identificador	RF-01
Actor involucrado	Usuario de tipo Cliente y Empresa
Nombre	Registro
Descripción	Como hay dos tipos de usuarios involucrados cada uno de ellos deberá rellenar un formulario distinto para poder registrarse en la plataforma. Una vez registrado en la plataforma podrá usar el correo y la contraseña introducidos para hacer login
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 15. Requisito funcional 1

Identificador	RF-02
Actor involucrado	Cualquier tipo de usuario
Nombre	Login
Descripción	El usuario que previamente se ha registrado en la plataforma podrá acceder a todas las funcionalidades de la herramienta introduciendo su correo y contraseña en el formulario de login. Una vez se haya realizado con éxito el login se mantendrá almacenada la información de sesión todo el rato que hagamos uso de la aplicación.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 16. Requisito funcional 2

Identificador	RF-03
Actor involucrado	Cualquier tipo de usuario de la plataforma
Nombre	Logout
Descripción	El usuario decide cerrar la sesión porque ya no va a hacer más uso de la plataforma o la duración de la sesión excede de un límite establecido. En el momento que se cumple una de las dos condiciones anteriores se elimina la información almacenada que hace referencia a la sesión, esto implica que si un usuario quiere acceder a todas las funcionalidades de la plataforma deberá volver a realizar el login.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 17. Requisito funcional 3

Identificador	RF-04
Actor involucrado	Cualquier tipo de usuario (Cliente, Empresa, Administrador)
Nombre	Buscar producto
Descripción	Cualquier tipo de usuario introduce unos criterios de búsqueda como puede ser el nombre del producto, el tipo de producto, la empresa que lo suministra... El sistema deberá mostrar un listado que cumpla los criterios de búsqueda que haya introducido el usuario. En función del tipo de usuario que vaya a realizar la búsqueda tendrá unos criterios diferentes.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 18. Requisito funcional 4

Identificador	RF-05
Actor involucrado	Empresa
Nombre	Subir imagen
Descripción	El sistema tiene que permitir subir imágenes para asociarlas a un producto que se va a ofertar en la plataforma. En el caso de que la empresa decida no subir ninguna imagen el sistema asignará una por defecto para mantener la estética en la plataforma.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 19. Requisito funcional 5

Identificador	RF-06
Actor involucrado	Empresa y Cliente
Nombre	Imprimir reserva
Descripción	Un usuario de tipo Empresa o Cliente podrá imprimir la información de la reserva en formato PDF, para facilitar la gestión de las reservas.
Prioridad	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 20. Requisito funcional 6

Identificador	RF-07
Actor involucrado	Empresa
Nombre	Imprimir listado de reservas
Descripción	El usuario de tipo empresa introducirá un periodo contenido entre dos fechas y el sistema imprimirá en formato PDF un listado con todas las reservas realizadas y el estado en el que se encuentran.
Prioridad	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 21. Requisito funcional 7

Identificador	RF-08
Actor involucrado	Cliente
Nombre	Realizar comentario/valoración
Descripción	Un usuario de tipo Cliente que previamente se ha identificado en la plataforma y está visualizando un producto. Este usuario rellenará el formulario para realizar comentario que además de contener un título y un texto dispondrá de un selector de estrellas para indicar lo satisfecho que está el usuario con el producto / trato de la empresa de alquiler. En el momento que se realiza una valoración el sistema calculará la valoración media del producto y se almacenará en la tabla de productos.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 22. Requisito funcional 8.

Identificador	RF-09
Actor involucrado	Cliente
Nombre	Calcular precio reserva
Descripción	Un usuario cliente se encuentra está visualizando un producto e introduce las fechas de inicio y fin para realizar una reserva. El sistema deberá calcular el precio total de la reserva a partir del precio por día del producto que está visualizando el usuario.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 23. Requisito funcional 9

Identificador	RF-10
Actor involucrado	Cliente
Nombre	Reservar un producto.
Descripción	Un usuario cliente se encuentra visualizando un producto e introduce las fechas de inicio y fin para realizar una reserva. A continuación, el sistema muestra el total de la reserva y si el usuario acepta ese precio, ese producto quedará reservado para ese cliente en las fechas indicadas.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 24. Requisito funcional 10

Identificador	RF-11
Actor involucrado	Cliente
Nombre	Obtener resultados por proximidad
Descripción	Un usuario cliente introduce su ubicación y realiza una búsqueda acorde a RF-04. El sistema deberá mostrarle los resultados ordenados por proximidad, es decir, en primer lugar, aparecerán los resultados que se encuentren más cerca de la ubicación introducida por el usuario
Prioridad	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Tabla 25. Requisito funcional 11

Identificador	RF-12
Actor involucrado	Administrador
Nombre	Ver usuarios de la plataforma
Descripción	El administrador accede a un apartado donde podrá visualizar un listado con los usuarios que se han registrado en la plataforma, tanto los de Tipo empresa como los de tipo Cliente.
Prioridad	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 26: Requisito funciona 12

Identificador	RF-13
Actor involucrado	Administrador
Nombre	Filtrar usuarios

Descripción	El administrador dispondrá de una serie de filtros que le permitirá localizar más fácilmente a un usuario de la plataforma, ya sea, Cliente o Empresa
Prioridad	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 27: Requisito funcional 13

Identificador	RF-14
Actor involucrado	Cualquier tipo de usuario identificado de la plataforma
Nombre	Visualizar valoraciones de un producto
Descripción	Cualquier usuario de la plataforma podrá visualizar las valoraciones de un producto ya que están ofertados en la plataforma.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 28: Requisito funcional 14

Identificador	RF-15
Actor involucrado	Empresa, Admin
Nombre	Calcular comisión
Descripción	El sistema calculará mediante un porcentaje fijo la comisión que deberá pagar una empresa por cada transacción que se realice dentro de la plataforma
Prioridad	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Tabla 29: Requisito funcional 15

Identificador	RF-16
Actor involucrado	Administrador
Nombre	Ver listado de comisiones
Descripción	El administrador tendrá un apartado donde podrá visualizar un listado con todas las comisiones generadas por las reservas realizadas por la plataforma. Este listado dispondrá de una serie de filtros para facilitar el proceso de clasificación
Prioridad	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Tabla 30: Requisito funcional 16

Identificador	RF-17
Actor involucrado	Empresa
Nombre	Elegir periodo de pago de comisión
Descripción	La empresa podrá decidir el periodo de pago de las comisiones generadas por cada transacción que se realice a través de EasyRent. Dispondrá de 3 opciones (mensual, trimestral, anual)
Prioridad	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/Opcional

Tabla 31: Requisito funcional 18

Identificador	RF-18
Actor involucrado	Empresa, Cliente
Nombre	Ver empresa en un mapa
Descripción	La empresa introducirá una dirección para poder disponer de una vista geolocalizada de la empresa a partir de sus datos.
Prioridad	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 32. Requisito funcional 18

6.4. Requisitos no funcionales.

Los requisitos no funcionales en cambio no describen el modo de actuar del sistema, sino que definen aspectos que se esperan del sistema, pero no describen funcionalidad en sí. Entre ellos se abordan aspectos referentes a la optimización a la seguridad etc.

A continuación, se describen una serie de requisitos no funcionales para la plataforma.

Identificador	RNF-01
Nombre	Optimización del tráfico de información
Descripción	Las consultas a las posibles API que se empleen para el funcionamiento de la plataforma serán mínimas para evitar tiempos de espera demasiado largos. Las consultas a la API serán mínimas, así como las llamadas a la BD para evitar tiempos de esperas largos
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 33. Requisito no funcional 1

Identificador	RNF-02
Nombre	Seguridad
Descripción	Aunque no son muchos, los datos que maneja la aplicación deben de cumplir con las normas mínimas de seguridad y protección de datos. Además de esto la comunicación del cliente con el servidor deberá realizarse bajo el protocolo de seguridad SSL (https).
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 34. Requisito no funcional 2

Identificador	RNF-03
Nombre	Multiplataforma
Descripción	El sistema debe de ser multiplataforma, y por tanto funcionar correctamente en los siguientes dispositivos: Andoid, iOS, Windows y Mac OS X. Además de esto la interfaz debe de ser responsive para poder visualizarla correctamente en las plataformas anteriormente mencionadas.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 35. Requisito no funcional 3

Identificador	RNF-04
Nombre	Escalabilidad
Descripción	El sistema se debe desarrollar de manera que en un futuro pueda ser escalable de manera sencilla. Para ello por ejemplo descargaremos responsabilidad sobre los dispositivos clientes.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 36. Requisito no funcional 4

Identificador	RNF-04
Nombre	Disponibilidad
Descripción	El sistema debe poder ser utilizado las 24 horas del día los 365 días del año, ya que, la API propia será alojada en un servidor que estará activo todo el tiempo.
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

Tabla 37. Requisito no funcional 5

7. Diseño

Una vez analizado los problemas y las necesidades relacionadas con el alquiler de ayudas técnicas podemos empezar a especificar y detallar los aspectos relativos al diseño de la propuesta que tiene la intención de resolver la problemática anteriormente comentada. A lo largo de esta sección se van a tratar todos los aspectos relativos al diseño de EasyRent tanto a nivel arquitectónico como a nivel visual.

A lo largo de este apartado vamos a diseñar y contemplar en primer lugar a EasyRent de manera global estableciendo una arquitectura conceptual y tecnológica. Una vez realizado este diseño distinguiremos dos secciones muy importantes. Por un lado, todo lo relativo al diseño del **back-end** como puede ser la persistencia o la API Rest y por otro lado toda la parte visual de la herramienta o **front-end** como pueden ser las interfaces o la paleta de colores empleada.

7.1. Diseño de la arquitectura conceptual

La arquitectura conceptual nos permite definir a grandes rasgos los módulos y bloques funcionales que va a contener EasyRent. Esta arquitectura es completamente independiente a los elementos tecnológicos que se emplearán para llevar a cabo este diseño, ya que, define aspectos arquitectónicos del sistema. Como EasyRent se trata de una plataforma web se rige bajo la arquitectura cliente-servidor.

La arquitectura **cliente-servidor** es un modelo de diseño de software donde las tareas se reparten entre los proveedores de recursos o servicios y los demandantes, es decir los llamados clientes realizan peticiones a través de internet mediante el uso un protocolo de comunicación que en la mayoría de los casos se emplea HTTP (*Hypertext Transfer Protocol*) y el servidor es el que recibe dicha petición la cual procesa y envía una respuesta. [16]

A continuación, podemos observar un diagrama que hace referencia a la arquitectura conceptual de EasyRent que también refleja la arquitectura anteriormente descrita:

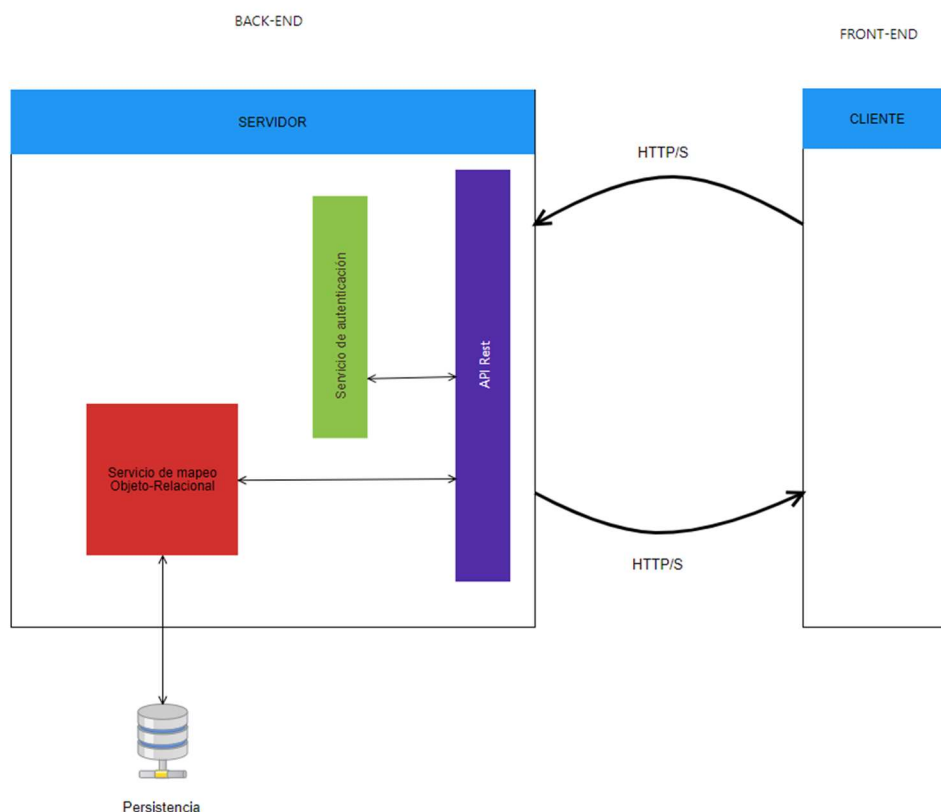


Figura 17: Diseño conceptual EasyRent. Fuente propia

Como podemos observar en la imagen anterior el cliente realiza peticiones a través del navegador web a la API Rest que mediante el servicio de autenticación verifica el usuario que realiza la dicha petición, una vez es validado, hace uso de un servicio de mapeo objeto-relacional para acceder a la información almacenada en la base de datos.

7.2. Diseño de la arquitectura tecnológica.

La arquitectura tecnológica nos permite especificar los elementos tecnológicos que se van a emplear para llevar a cabo el diseño conceptual comentado anteriormente. La elección de los elementos tecnológicos necesarios para llevar a cabo un proyecto de desarrollo software está condicionada o una serie de factores:

- Las necesidades y requerimientos del proyecto.
- La experiencia previa de los desarrolladores con ciertos elementos tecnológicos
- O simplemente puede venir impuesto por un tercero.

En primer lugar, vamos a especificar los elementos tecnológicos que hacen referencia a la parte del servidor o back-end.

Para la persistencia de datos se ha decidido hacer uso de bases de datos relacionales *MySQL*, ya que este tipo de sistemas de almacenamiento de información se adecua perfectamente a los requerimientos del proyecto, en los siguientes apartados comentaremos más detalles acerca del modelo de bases de datos que se va a emplear.

Por otro lado, para la realización de la API Rest se va a utilizar el framework escrito en JavaScript Express bajo el entorno de programación y ejecución *NodeJs*, ya que, permite implementaciones rápidas y funcionales de servicios web, además este entorno dispone de un manejador de paquetes *npm* que nos permite emplear software de terceros de manera rápida y sencilla, cabe destacar que he escogido este framework debido a que dispongo de cierta experiencia con esta tecnología al haberla empleado en desarrollos anteriores.

Como he mencionado anteriormente *NodeJS* dispone de un manejador de paquetes que nos brinda la posibilidad de emplear software de terceros, en este caso tanto el servicio de autenticación como el servicio de mapeo de objetos con la base de datos se ha extraído de *npm* haciendo uso del ORM (*Object-Relational Mapping*) *Sequelize* para el mapeo de la base de datos y de la librería *Passport* para crear el servicio de autenticación basado en JWT (JSON web Tokens) que comentaremos más adelante.

En segundo lugar, vamos a especificar los elementos tecnológicos empleados para la implementación del front-end o parte visual del proyecto.

Para desarrollar las interfaces de usuario he decidido usar uno de los frameworks más empleados actualmente como es el caso de *Angular* en su versión 7.2 junto a *Angular Material* como framework de CSS para facilitar la implementación gracias a los componentes de gran calidad visual de los que dispone.

Por último, en cuanto a los elementos tecnológicos cabe destacar la integración de la API de *Google Maps* para situar en el mapa a las empresas que decidan hacer uso de EasyRent.

En la imagen siguiente a modo de resumen podemos observar todos los elementos tecnológicos anteriormente mencionados.

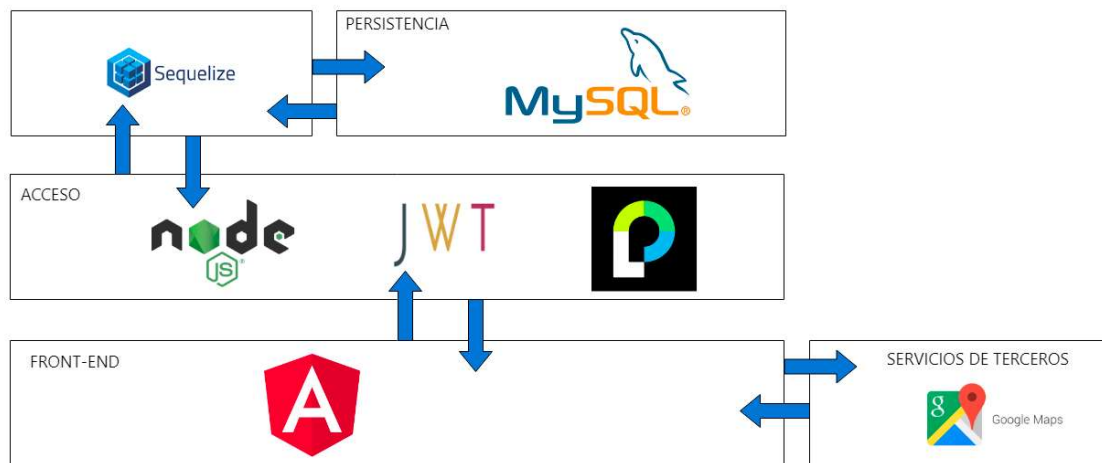


Figura 18: Stack tecnológico EasyRent. Fuente propia

7.3. Diseño de los aspectos relativos al back-end.

A lo largo de esta sección vamos a entrar en más detalles en los elementos tecnológicos y arquitectónicos relativos al back-end como son la persistencia de datos, el diseño de la API Rest y algunos aspectos de seguridad.

7.3.1. Persistencia de datos.

Dado que EasyRent se entiende como un gestor de reservas y productos es necesario almacenar y tener disponible la información en todo momento para ello es necesario disponer de una base de datos relacional.

Las entidades más importantes de la base de datos de EasyRent son:

- Usuarios: Esta tabla es la encargada de almacenar la información relativa a los usuarios que se registran en la plataforma mediante un correo y una contraseña. Los campos que presenta esta tabla son los siguientes:
 - *ID*: Clave primaria y de tipo autoincremento.
 - *Nombre*: tipo varchar2 para almacenar el nombre del usuario.
 - *Email*: tipo varchar2 y con restricción de valor único para evitar que haya dos usuarios con el mismo email.
 - *Password*: tipo varchar2 para almacenar la contraseña cifrada.
 - *FotoPerfil*: tipo varchar2 que nos permite almacenar la ruta donde se almacenará la foto de perfil por defecto se asigna una imagen preestablecida.
 - *IdTipo*: Clave ajena hacia la tabla Tipo_Usuario que nos permite diferenciar el tipo de usuario (Cliente, Admin, Empresa).
- Empresas: Esta tabla se basa en el concepto de restricción de identificación de las bases de datos relacionales, es decir, la clave primaria de esta tabla es a su vez clave ajena hacia la tabla Usuarios. Esta técnica nos permite realizar una especificación del tipo de usuario añadiendo campos a un solo tipo de usuario, en este caso las empresas. Por

tanto, una Empresa contendrá todos los datos de un usuario normal más unos específicos para la empresa. Los campos específicos de esta tabla son:

- *Descripción*: de tipo varchar2 para almacenar una descripción de la empresa para informar al usuario.
- *IdLugar*: Clave ajena hacia la tabla lugares para indicar la dirección de la empresa donde se recogerán los productos, en este caso ayudas técnicas
- Productos: Esta entidad permite conservar la información de los productos que se van a ofertar en la plataforma y tenerlos asociados a una empresa. Esta tabla tendrá los siguientes campos:
 - *ID*: Clave primaria de tipo autoincrementante.
 - *Nombre*: de tipo varchar2 para almacenar el nombre del producto.
 - *ShortDescription* y *Descripcion*: Ambos de campos de tipo varchar2 para almacenar dos descripciones una más larga que la otra.
 - *Precio*: de tipo decimal para almacenar la información del precio del producto.
 - *IdEmpresa*: clave ajena hacia la tabla Empresas anteriormente comentada.
 - *Valoración*: indica la valoración media del producto.
 - *Tipo*: En una primera implementación será un valor numérico que indique si el producto es manual o dispone de algún tipo de motor o mecanismo que lo haga automático.
- Valoraciones: El objetivo de esta entidad es almacenar las valoraciones que realicen los usuarios de la plataforma sobre cada uno de los productos. Para ello esta tabla contendrá los siguientes campos:
 - *Título y Texto*: de tipo varchar2 para almacenar la información básica de la valoración.
 - *Estrellas*: valor numérico para representar la cantidad de estrellas que se le asignan al producto sobre el que se está haciendo la valoración.
 - *IdUsuario, IdProducto*: Claves ajenas hacia las tablas Usuarios y Productos
- Imágenes: La función principal de esta tabla es guardar la ruta donde se almacenan las imágenes asociadas a un producto concreto. Para lograr esto se ha empleado una relación 1 a N es decir un producto puede tener muchas fotos y una foto solo puede ser

de un producto únicamente debemos incluir una clave ajena en la tabla imágenes que apunte hacia la tabla productos.

- Reservas: Esta entidad es la encargada de almacenar toda la información relativa a la reserva que se realicen dentro de la plataforma para ello deberá contener los siguientes campos:
 - *FechaInicio y FechaFin*: de tipo Data para establecer la duración de la reserva.
 - *Precio*: de tipo Decimal para establecer el precio de la reserva en función de la duración y del precio del producto.
 - *IdEstado*: clave ajena hacia la tabla estados para saber en qué estado está la reserva,
 - *IdUsuario, IdProducto*: claves ajenas hacia las tablas de usuarios y productos para identificar el usuario y el producto implicado en la reserva.
- Lugares: Esta tabla será la encargada en almacenar la información obtenida a partir de la API de Google Maps para saber las direcciones de las empresas. Los campos de esta tabla son los siguientes, aunque están sujetos a variaciones en función de la información que se extraiga de la API de Google Maps:
 - *Id*: identificador proporcionado por la API.
 - *Dirección*: para almacenar el nombre de la dirección.
 - *Latitud, longitud*: para poder situar las direcciones en un mapa.
- Comisiones: Esta entidad va a ser la encargada de almacenar todas las comisiones generadas por cada una de las reservas que se realicen en la plataforma. Para lograr esto la tabla deberá contener los siguientes campos:
 - *Comisión*: de tipo decimal para almacenar la comisión generada por el precio de una reserva y un porcentaje preestablecido.
 - *IdReserva*: clave ajena hacia la tabla reserva para obtener la información de la reserva.

Cabe destacar que todas las entidades anteriores además de los campos comentados anteriormente tienen los campos *createdAt* y *updatedAt* debido a que son necesarios para la utilización del ORM Sequelize que nos permite mapear la base de datos de una manera rápida y sencilla. En el diseño de la API Rest comentaremos más detalles al respecto.

Por último, entidades Tipo_Pago, Tipo_Usuario, Estado_Reserva y Estado_Producto de la base de datos siguen la misma estructura ya que se tratan de enumeraciones para almacenar valores que no van a sufrir modificaciones y se emplean para establecer estados, tipos ... Las tres tablas

tendrán un valor numérico a modo de clave primaria de tipo autoincremento y un nombre para identificar de manera visual el valor.

A continuación, podemos observar el modelo entidad relación para entender de manera visual todo lo comentado anteriormente.

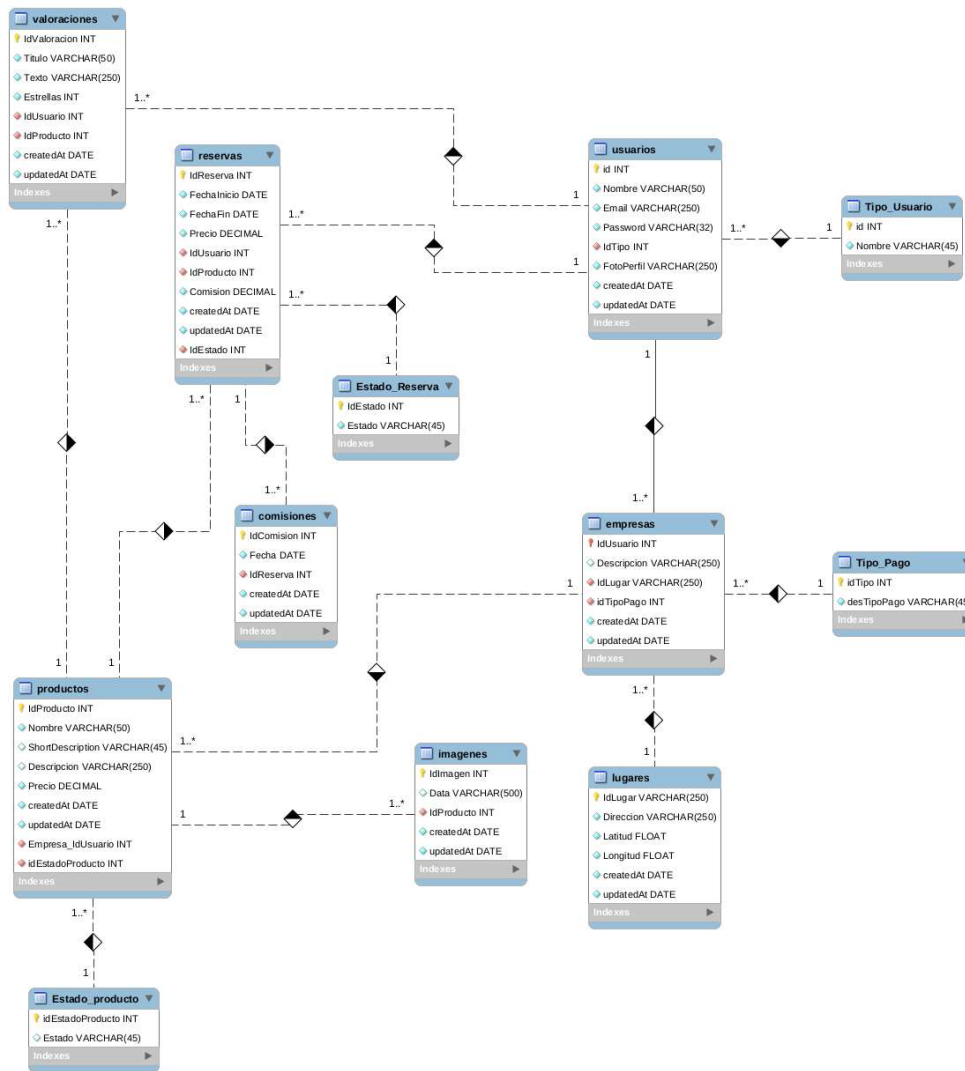


Figura 19: Modelo entidad relación de EasyRent. Fuente propia

7.3.2. Diseño API

A lo largo de este apartado se van a comentar aspectos de diseño tanto arquitectónico como una especificación de las rutas disponibles de nuestra API Rest para comunicar el front-end con el back-end.

En cuanto al nivel arquitectónico la Api Rest está basada en el patrón MVC (Modelo, Vista, Controlador) pero se le añaden dos componentes o middlewares que hace referencia a por un lado a las rutas que permiten la comunicación del front-end con la API Rest estableciendo los end-points y por otro a los aspectos de autenticación. Cabe destacar que en este caso la Vista es la parte visual donde del front-end desarrollada en angular por lo que entraremos en detalle más adelante.

Por tanto, el flujo de datos en la API Rest será el siguiente: en primer lugar, la ruta recoge la información sobre la petición que se envía al componente de autenticación que en caso de ser correcta dicha identificación continua el proceso enviando los datos al controlador que es el encargado de realizar toda la lógica de negocio y en caso de que necesite acceder o modificar la información de a base de datos lo hará a través de un modelo previamente definido con *Sequelize*.

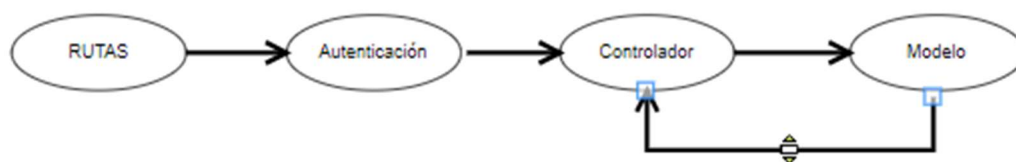


Figura 20: Flujo de datos API Rest. Fuente propia

Una vez el controlador termina de realizar todas las operaciones construye una respuesta en formato JSON con un formato estándar para cada una de las rutas que se establezcan, esta estructura estará compuesta por un campo *success* que puede tomar valor *true* o *false* que indica si se ha realizado correctamente la petición, un campo *message* para indicar una pequeña descripción de lo sucedido y un campo *data* que contendrá la información solicitada. A continuación, podemos ver un ejemplo de esto.

```
{
  "success": true,
  "message": "Login correcto",
  "data": {
    "user": {},
    "token": {}
  }
}
```

Figura 21. Ejemplo de respuesta de la API. Fuente propia

Por último, se detallan todas las rutas que se van a hacer uso para comunicar el front-end con el back-end.

Ruta	/usuario/:id
Autorización	Cliente
Método	GET
Parámetros	Id de un usuario
Cuerpo	-
Descripción	Devuelve la información del usuario con el id pasado como parámetro

Ruta	/empresa/:id
Autorización	Cliente
Método	GET
Parámetros	Id de un usuario de tipo empresa
Cuerpo	-
Descripción	Devuelve toda la información del usuario pasado como parámetro, así como la información propia de la empresa.

Ruta	/usuario/
Autorización	
Método	POST
Parámetros	
Cuerpo	Recibe un objeto JSON con los campos Nombre, Password, Email.
Descripción	Esta ruta es la encargada de dar de alta un usuario en la plataforma para que posteriormente pueda hacer login.

Ruta	/usuario/empresa
Autorización	
Método	POST
Parámetros	
Cuerpo	Recibe un objeto JSON con los campos necesarios para dar de alta a un usuario y a una empresa

Descripción	Esta ruta es la encargada de dar de alta un usuario de tipo empresa en la plataforma para que posteriormente pueda iniciar sesión en la plataforma.
-------------	---

Ruta	/usuario/user
Autorización	Cliente
Método	PUT
Parámetros	
Cuerpo	Recibe un objeto JSON con todos los datos del usuario que desea modificar.
Descripción	Esta ruta permite a todos los tipos de usuario actualizar la información básica como puede ser el nombre, la contraseña, la foto de perfil...

Ruta	/usuario/empresa
Autorización	Empresa
Método	PUT
Parámetros	
Cuerpo	Recibe un objeto JSON con todos los datos específicos de la empresa.
Descripción	Esta ruta permite a un usuario de tipo Empresa actualizar los datos propios y relativos a una empresa como pueden ser la dirección, la descripción o el tipo de pago

NOTA: Se ha decidido separar en dos rutas la parte de actualización de los usuarios para simplificar el proceso de modificación.

Ruta	usuario/login/
Autorización	
Método	POST
Parámetros	
Cuerpo	Recibe un objeto JSON con los campos Email y Password
Descripción	Método encargado de realizar el login en la plataforma en el caso de que el login sea correcto devuelve información sobre el usuario y el token de acceso.

Ruta	/productos/find/p={pag}
Autorización	-

Método	POST
Parámetros	El número de página en el que se encuentra
Cuerpo	Un JSON con los criterios de búsqueda necesarios para filtrar la información
Descripción	Devuelve un listado de productos filtrados por el objeto JSON incluido en el cuerpo de la petición. Haciendo uso del join muestra información del producto en sí, las imágenes asociadas a ese producto y la empresa a la que pertenece.

Ruta	/productos/:id
Autorización	-
Método	GET
Parámetros	Id de un producto
Cuerpo	-
Descripción	Devuelve toda la información de un producto con el id pasado por parámetro. Gracias Sequelize con una única petición y una consulta a la base de datos podemos obtener la información de las valoraciones que tiene, la empresa a la que pertenece y las imágenes asociadas al producto

Ruta	/productos
Autorización	Empresa
Método	POST
Parámetros	
Cuerpo	Recibe un objeto JSON con los campos necesarios para crear un producto. Nombre, Precio, ShortDescripcion, Descripcion, IdEmpresa.
Descripción	Permite dar de alta los productos que se van a ofertar en la plataforma.

Ruta	/productos
Autorización	Empresa
Método	PUT
Parámetros	
Cuerpo	Recibe un JSON con los campos para modificar de un producto.

Descripción	Permite modificar un producto dado de alta en la plataforma. Esta petición solo la podrá realizar un usuario de tipo empresa.
-------------	---

Ruta	/productos/{id}
Autorización	Empresa
Método	DELETE
Parámetros	El id del producto que se desea eliminar
Cuerpo	
Descripción	Permite eliminar un producto de la plataforma. Esta petición únicamente la podrán realizar los usuarios de tipo Empresa

Ruta	/reserva/{id}
Autorización	Usuario
Método	GET
Parámetros	El id de la reserva a consultar
Cuerpo	
Descripción	Obtiene toda la información relativa a la reserva pasada por parámetro. Obtiene la información del producto asociado a la reserva y la información de la empresa a la que pertenece dicho producto

Ruta	/reserva/print/{id}
Autorización	Usuario
Método	GET
Parámetros	El id de la reserva a imprimir
Cuerpo	
Descripción	Esta petición genera un archivo PDF con toda la información relativa a la reserva pasada por parámetro. Una vez generado el archivo se visualizará en el navegador del cliente dando la opción a descargar el archivo.

Ruta	/find/{p=pag}
Autorización	
Método	POST

Parámetros	La página solicitada con los criterios de búsqueda establecidos
Cuerpo	Un objeto JSON con el id del usuario que solicita la búsqueda y los criterios de búsqueda introducidos por el usuario, estos criterios podrán ser fecha inicial y final, estado de la reserva...
Descripción	Obtiene un listado con la información de las reservas realizadas por un usuario logeado atendiendo a los criterios de búsqueda establecidos.

Ruta	/reserva/
Autorización	Usuario
Método	POST
Parámetros	
Cuerpo	Un objeto JSON con los campos necesarios para realizar una reserva
Descripción	Esta petición es la encargada de realizar la reserva e introducirla en el sistema. Cabe destacar que esta petición es la encargada de realizar validaciones de gestión para no permitir por ejemplo reservar un producto por dos usuarios distintos en la misma fecha.

Ruta	/valoracion/
Autorización	Usuario
Método	POST
Parámetros	
Cuerpo	Un objeto JSON con la información necesaria para realizar una valoración dentro del sistema
Descripción	Esta petición permite a los usuarios logeados en la plataforma dejar sus opiniones y/o valoraciones sobre un producto ofertado en la plataforma.

Ruta	/valoracion/{idProducto}
Autorización	Usuario
Método	GET
Parámetros	El id del producto
Cuerpo	

Descripción	Esta petición obtiene todas las valoraciones asociadas al producto pasado por parámetro.
-------------	--

Ruta	/valoracion/{idValoracion}
Autorización	Usuario
Método	DELETE
Parámetros	El id de la valoración
Cuerpo	
Descripción	Esta petición permite a los usuarios que propietarios de la valoración eliminarla de la plataforma.

7.3.3. Aspectos de seguridad.

A lo largo de este apartado se van a comentar algunos aspectos relativos con la seguridad de la plataforma tanto a nivel de comunicación como a nivel de la API o a la hora de almacenar información en la base de datos.

En primer lugar, haremos uso del protocolo de comunicación HTTPS cifrado para realizar conexiones entre el front-end y la API Rest por un canal seguro. Para lograr esto es necesario disponer de certificados digitales firmados por una autoridad certificadora, en este caso haremos uso de los certificados auto firmados distribuidos por *Let's Encrypt*.

Por otro lado, me gustaría destacar que el cifrado empleado para almacenar las contraseñas de los usuarios que hacen uso de la plataforma se emplea la función de cifrado *bcrypt* que hace uso del algoritmo de cifrado *blowfish*. Que genera un hash único para una entrada, la fortaleza de este algoritmo de cifrado reside en que el proceso es unidireccional, es decir, a partir de un texto plano es sencillo generar su correspondiente hash, pero a la inversa es computacionalmente muy costoso por lo que este algoritmo es muy utilizado para almacenar contraseñas.

Por último, me gustaría comentar que gracias al middleware de npm Passport nos permite establecer diferentes estrategias de seguridad, ya sea empleando JSON Web Tokens, servicios OAuth...

En el caso de la API Rest de EasyRent se establecen estrategias para un usuario Cliente, Administrador o Empresa, para restringir rutas concretas a un tipo de usuario.

7.4. Diseño de los aspectos relativos al front-end.

A lo largo de esta sección se estudia la manera en la que se diseña EasyRent con el objetivo de proporcionar una interfaz clara y usable, para ello se aplicará el principio de “Interfaz autodescriptiva” y se planteará un diseño basado en el “Flat design”.

Posteriormente, se especificará la paleta de colores a emplear durante el desarrollo, proporcionando una guía de estilos con los bocetos y MockUps referentes a las diferentes pantallas de la aplicación.

7.4.1. Conceptos previos

7.4.1.1. Principio de interfaz autodescriptiva.

Por autodescriptivo entendemos que se describe a sí mismo, por tanto, una interfaz autodescriptiva es aquella donde los elementos que la componen tienen expresividad, de tal forma que el usuario pueda llegar a entender el funcionamiento y para qué sirve cada uno de los elementos presentes en la interfaz.

Para lograr esto estas interfaces hacen uso de una serie de elementos que son muy representativos a la acción que van a realizar, los más importantes son los siguientes:

- **Iconos:** Gracias a ellos podemos describir acciones que no requieren del conocimiento de un idioma, ya que, los iconos suelen representar acciones de la vida cotidiana o aspectos muy presentes en ella. Un ejemplo muy claro de esto es el uso del icono de una papelera para expresar la acción de borrar o eliminar.
- **Colores:** Al igual que los iconos los colores aportan y expresan información que no tiene nada que ver con el conocimiento de un idioma. Por ejemplo, un botón de color verde se emplea para confirmar cambios o acciones correctas.
- **Presentar elementos ocultos:** a partir de una determinada acción de un usuario, se muestran elementos que aportan significado o información adicional sobre la acción que se va a realizar, ejemplos de estos elementos ocultos pueden ser los *tooltips*, *popovers* o los conocidos *modales* o *popUps* de confirmación.
- **Animaciones:** Este tipo de elemento es más peculiar, ya que, si no se usan de manera correcta puede ocasionar el efecto contrario y hacer que el usuario se maree y no entienda la interfaz. Un ejemplo de un uso correcto y muy habitual es el uso de objetos intermitentes para insinuar el inicio de un proceso.

7.4.1.2. Principios del Flat design.

El *Flat design* o diseño plano consiste en eliminar o reducir todo tipo de elemento o decoración en un diseño de una interfaz para simplificar el mensajes y facilitar la comprensión de la funcionalidad por parte del usuario que va a emplearla, para ello se eliminan texturas, degradados, sombreados... en definitiva todo lo que no aporta valor al mensaje o información que se quiere transmitir al usuario que interactúa con la interfaz [17].

Además de esto hay que tener en cuenta aspectos relativos a los colores y a la tipografía. En cuanto a los colores el *Flat design* recomienda emplear pocos colores con pocas variaciones en cuanto a la tonalidad y hacer uso tipografías sans serif debido a que son las más sencillas para su lectura.

7.4.2. Diseño de interfaces.

Antes de comenzar a desarrollar una aplicación web es conveniente invertir un tiempo en realizar diseños preliminares o mockups de las interfaces que va a contener toda la funcionalidad del sistema, ya que, de esta manera por un lado los clientes pueden ver un prototipo visual de lo que desea y los desarrolladores inician el desarrollo desde un punto de partida lo cual facilita el proceso.

Cabe destacar que este diseño de interfaces debe de ser capaz de resolver y abordar todos los requisitos funcionales y no funcionales anteriormente descritos, también como hemos mencionado anteriormente deben estar basados en las interfaces auto descriptivas y cumplir los principios del *flat design*.

Estos diseños de interfaces se clasifican en función del nivel de fidelidad y de detalle que muestran:

- Bocetos: Representaciones rápidas normalmente en papel para realizar la primera aproximación de manera visual de la aplicación que se va a desarrollar
- Wireframe: Es la segunda representación de nuestra propuesta de solución, normalmente utiliza cajas y símbolos entendibles y está orientado a reflejar de una forma un poco refinada las funcionalidades del sistema y dónde encajan más o menos los elementos que lo forman. En este momento ya se pueden incluir los aspectos de

usabilidad y accesibilidad que están más relacionados con las funcionalidades del sistema. La principal ventaja de estos prototipos es que se realizan de manera muy rápida pero normalmente distan bastante del producto final

- Mockup: Es una representación visual **Error! No se encuentra el origen de la referencia.** exacta de cómo será la interfaz que percibirá el usuario y por tanto indica no solo la posición de los elementos sino su estado final una vez implementados, contemplando aspectos como los colores, los iconos empleados. La principal ventaja de este diseño es que ya es muy cercano al desarrollo, pero requiere un gasto de tiempo elevado, ya que, requiere aspectos puramente relacionados con el diseño.

7.4.3. Prototipos de EasyRent.

Como hemos mencionado anteriormente para realizar un buen diseño de interfaces es necesario realizar unos prototipos visuales que muestren el aspecto final de la solución propuesta.

Para EasyRent en primer lugar se han realizado una serie de bocetos en papel para contemplar las principales pantallas que va a presentar la plataforma, a continuación, podemos observar este primer acercamiento.

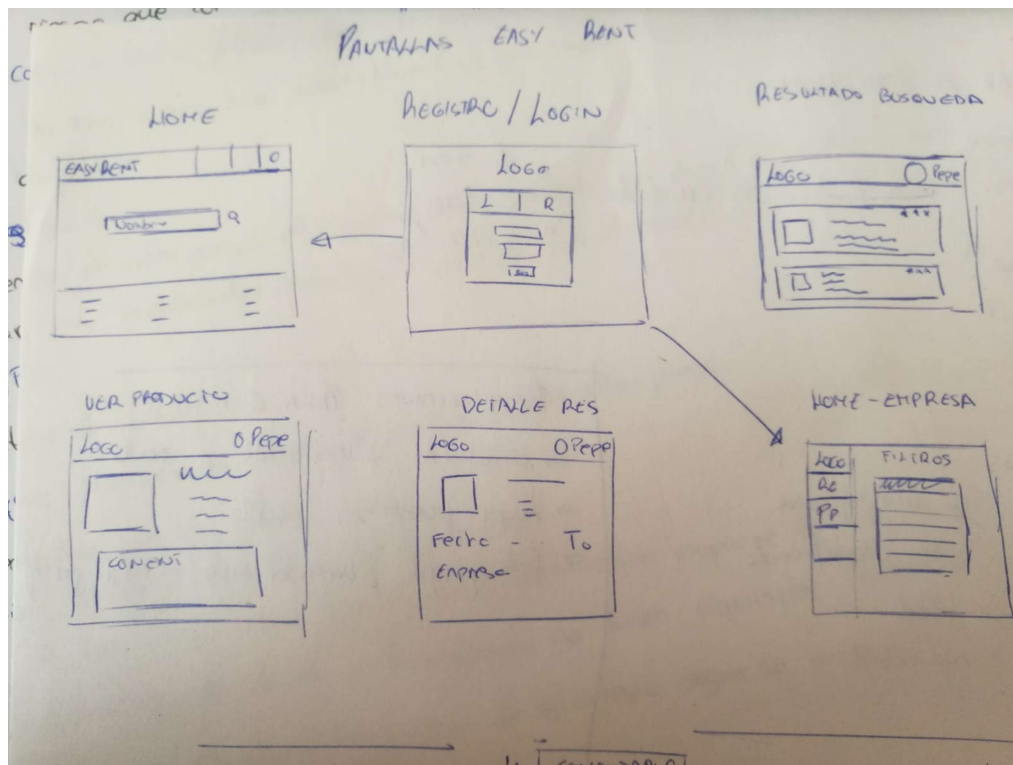


Figura 22: Bocetos EasyRent. Fuente propia

En este primer diseño podemos apreciar elementos clave como va a ser la disposición del menú superior, como se va a estructurar el listado de productos o como se va a mostrar la información de un producto concreto, cabe destacar que este primer diseño es muy básico ya que su objetivo es analizar y especificar las interfaces que va a tener el sistema sin entrar en demasiado detalle.

Este primer diseño nos permite conocer el esqueleto del sistema y por tanto nos va a permitir entrar en más detalles a la hora de realizar los Wireframes, debido a que previamente ya habíamos hecho el esfuerzo de especificar cierta información relativa a las interfaces.

Para realizar los Wireframes de EasyRent se ha empleado la herramienta online Balsamiq por su sencillez y la cantidad de elementos de los que dispone para realizar este tipo de representaciones. A continuación, podemos observar de manera global como será EasyRent:

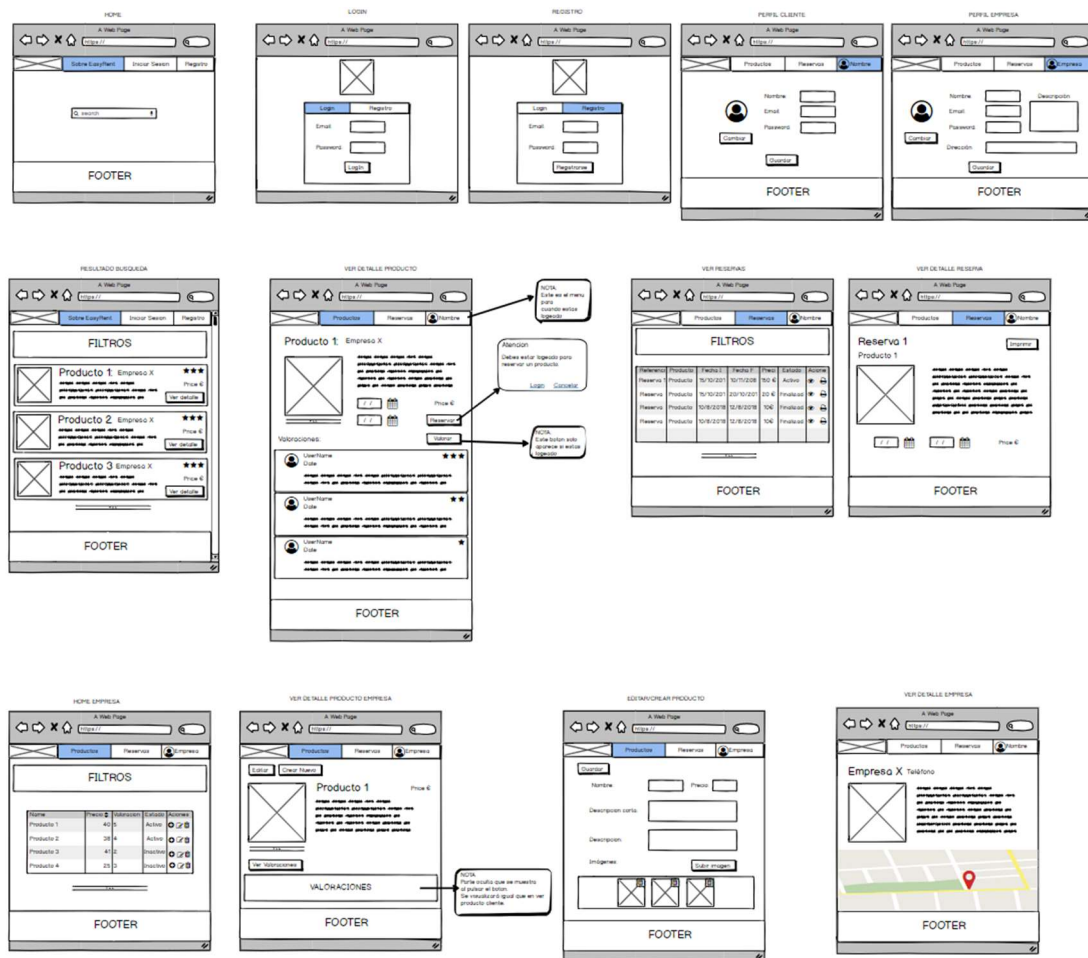


Figura 23: Wireframe global de EasyRent. Fuente propia

En este diseño podemos observar un aumento considerable en el número de interfaces, así como un incremento en el nivel de detalles de este, así como anotaciones de elementos o acciones que únicamente se podrán realizar en el caso de que se haya iniciado la sesión.

En la página principal podemos observar que estará formada por el menú superior con las opciones de *Iniciar sesión*, *regístrate*, *productos* y *Acerca de*. En esta home-page podemos observar que aparecerá un campo para poder realizar una búsqueda de los productos.

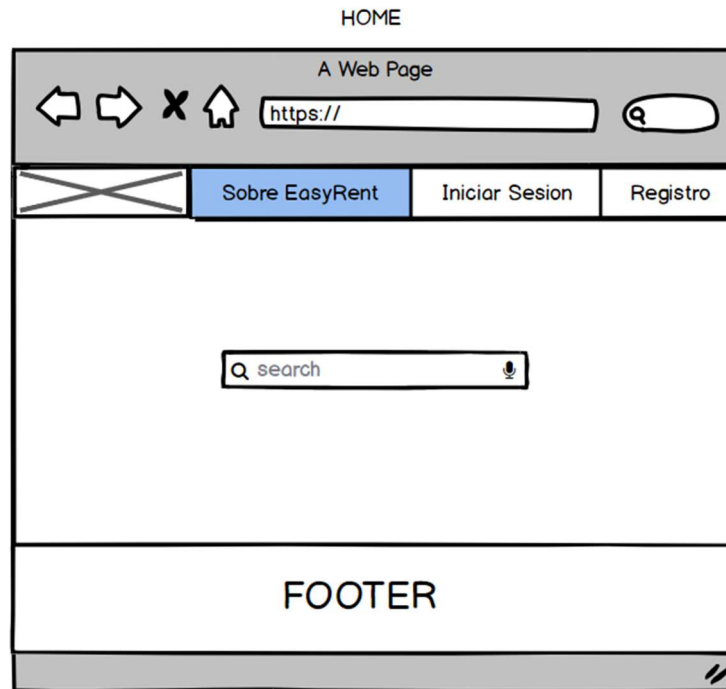


Figura 24. Wireframe Home-page. Fuente propia

El siguiente wireframe es uno de los más importantes, debido a que, va a ser la parte que más accesos reciba, ya que, muestra los resultados de una búsqueda y el detalle de un producto, por tanto, esta parte tiene que tender un diseño cuidado y atractivo.

En cuanto al resultado de una búsqueda podemos apreciar como en la parte superior se mantiene el menú y justo debajo los filtros disponibles para buscar un producto concreto, también cabe destacar el listado de productos donde ya queda claro la información que se va a mostrar, así como las medias de las valoraciones que tiene el producto. Por último, en la parte inferior podemos observar que los resultados irán paginados y algo información adicional en un footer.

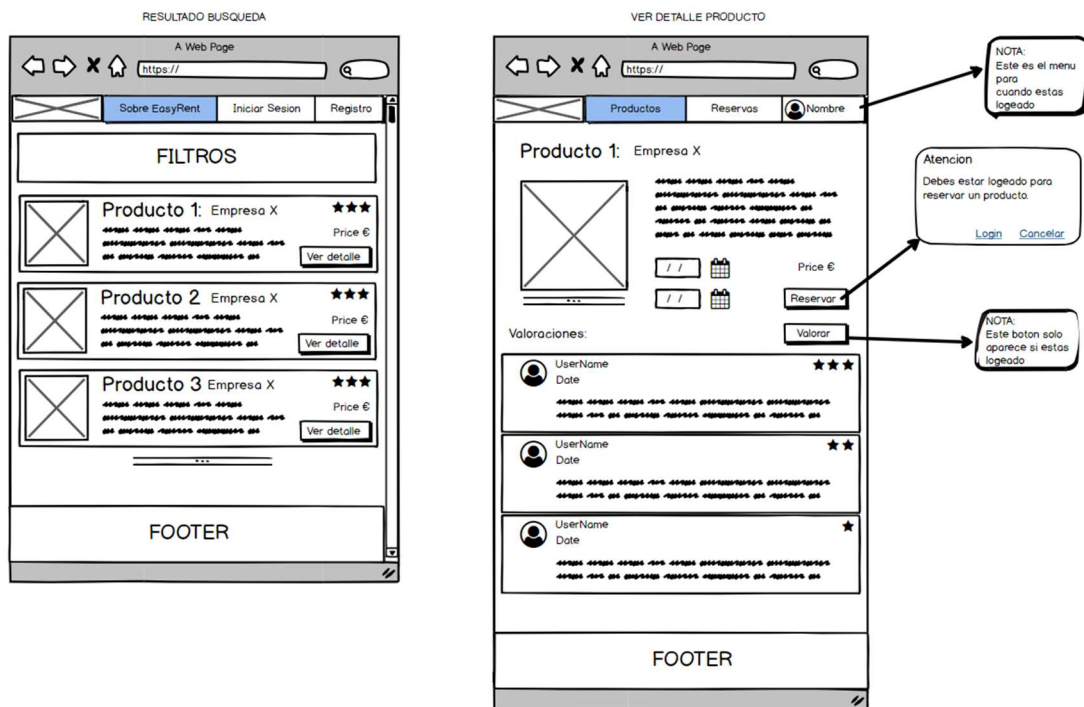


Figura 25. Resultado de búsqueda y detalle de un producto. Fuente propia

En cuanto a la visualización de un producto podemos observar una parte donde se muestra la información del producto, así como un carrusel con las imágenes asociadas a ese producto. También cabe destacar que en la parte inferior aparecen las valoraciones realizadas por los usuarios de la plataforma sobre el producto que se está visualizando. Estas valoraciones contradirán información sobre el usuario que la realiza, la fecha en la que se crea y la valoración en si, además de la puntuación de 0 a 5 estrellas.

EasyRent va a permitir a los usuarios registrados ver información sobre las reservas que realicen dentro de la plataforma por tanto es necesario tener una interfaz que atienda estos requisitos. El siguiente wireframe hace referencia a esta parte de la aplicación. En el lado izquierdo del wireframe podemos visualizar una tabla que contendrá una tabla con la información básica de una reserva, así como una columna con una serie de acciones rápidas. En cambio, en el lado derecho podemos observar el detalle de una única reserva con más información del producto que se ha reservado, así como las fechas en las que se ha reservado.

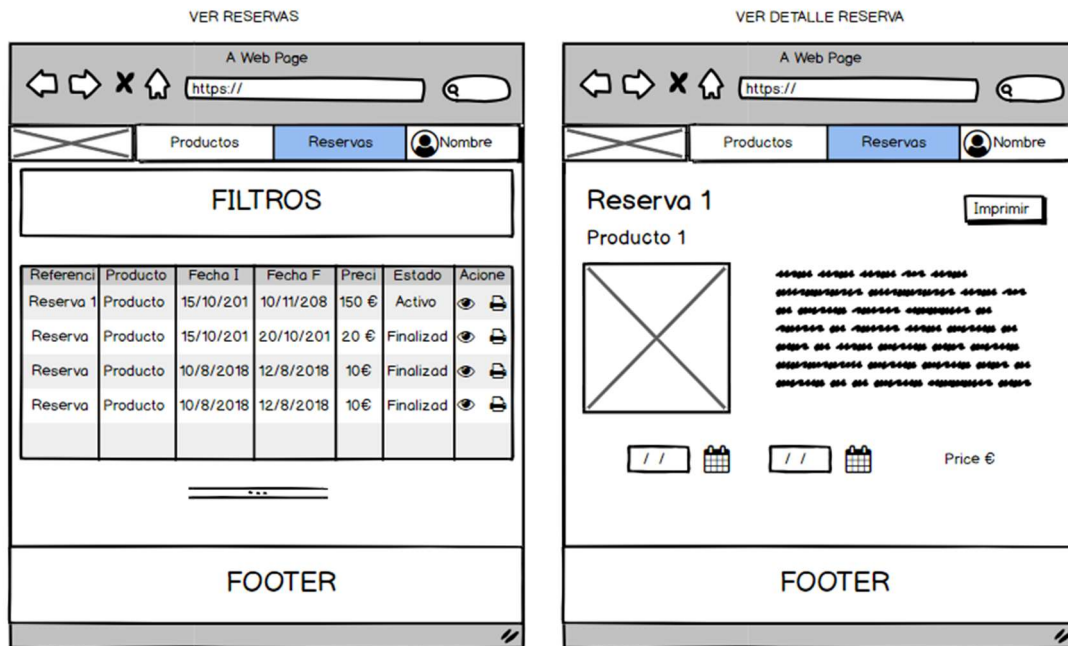


Figura 26. Ver reservas. Fuente propia

Como hemos mencionado en apartados anteriores las empresas que se registren en EasyRent podrán dar de alta sus productos para poder ofertarlos dentro de la plataforma como hemos visto en anteriores wireframes para ello es necesario diseñar una interfaz para que las empresas puedan visualizar los productos de manera clara.

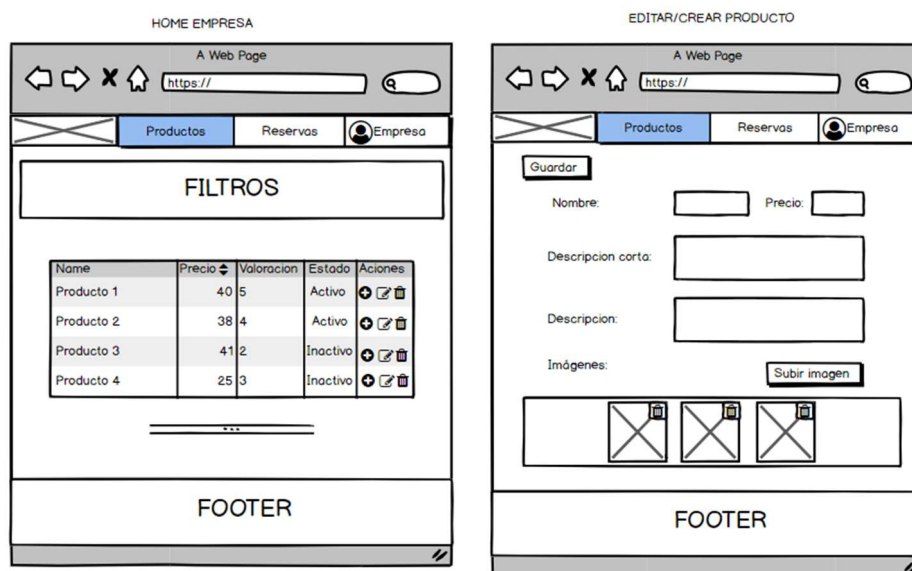


Figura 27. Wireframes para la gestión de productos. Fuente propia

En el wireframe de la izquierda siguiendo con la estética anterior podemos visualizar una tabla con la información básica de los productos de la empresa que haya iniciado sesión, así como un paginado para que el scroll no sea excesivo en el caso de que tenga muchos productos. En cambio, el wireframe de la derecha es el encargado de permitir la edición y la creación de nuevos productos, en el podemos observar un formulario con los campos necesarios, así como una sección para incluir las imágenes asociadas a este producto.

También cabe destacar que EasyRent va a permitir modificaciones en la información básica, tanto para los usuarios de tipo cliente como la de las empresas, para ello es necesario diseñar una interfaz que permita realizar estas acciones de manera rápida y cómoda.

The image displays two side-by-side wireframes for user profile management. The left wireframe is titled 'PERFIL CLIENTE' and the right is 'PERFIL EMPRESA'. Both are presented as browser windows with a title bar 'A Web Page' and a URL bar containing 'https://'. Each wireframe features a navigation bar with three tabs: 'Productos', 'Reservas', and a profile icon. In the 'PERFIL CLIENTE' version, the profile icon is labeled 'Nombre', while in the 'PERFIL EMPRESA' version, it is labeled 'Empresa'. The main content area of the 'PERFIL CLIENTE' wireframe includes a profile icon, input fields for 'Nombre:', 'Email:', and 'Password:', a 'Cambiar' button, and a 'Guardar' button. The 'PERFIL EMPRESA' wireframe includes a profile icon, input fields for 'Nombre:', 'Email:', 'Password:', and 'Dirección', a 'Descripción' text area, a 'Cambiar' button, and a 'Guardar' button. Both wireframes conclude with a 'FOOTER' section at the bottom.

Figura 28. Páginas de perfil EasyRent.fuente propia

En este caso ambas interfaces serán muy similares, ya que, tiene la misma funcionalidad únicamente cambia la información que van a poder modificar.

Por último, los usuarios que lo deseen podrán visualizar la información de una empresa que este ofertando sus productos en EasyRent para conocer más de ella o saber dónde tienen situado el punto de entrega/recogida etc. Para ello se ha diseñado una interfaz sencilla que incluye un mapa de Goolge Maps con un marcador en la posición que haya establecido la empresa. Esto lo podemos apreciar en la Figura 29. Ver detalle empresa. Fuente propia

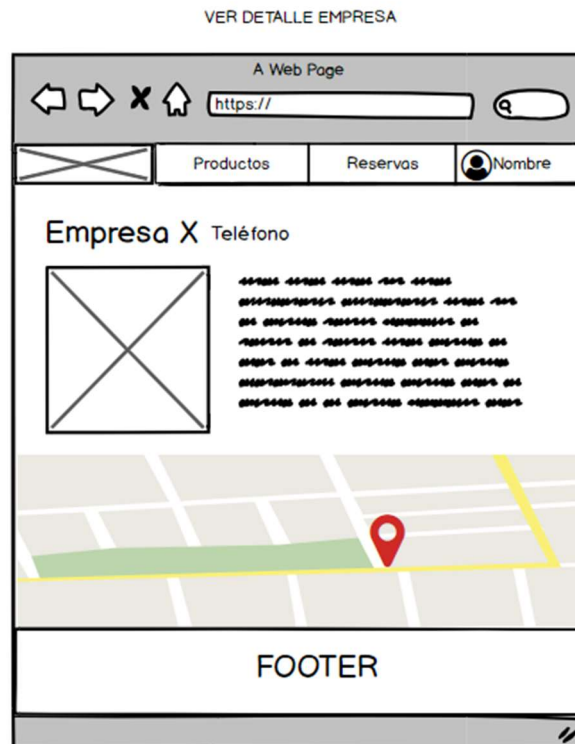


Figura 29. Ver detalle empresa. Fuente propia

7.4.4. Guías de estilo.

Una guía de estilo es un documento que tiene el objetivo de especificar un estilo o formato estándar de una aplicación. En ella se establecen las directivas comunes en cuando a diseño de elementos, fuentes a utilizar, formato de las imágenes, colores, etc. El uso de este tipo de documentos en el ámbito empresarial es clave ya que proporciona dos beneficios clave a la hora del desarrollo de la aplicación:

- Por un lado, la posibilidad de presentar al cliente, superior o supervisor un documento, donde podrá ver de manera clara como se van a diseñar los elementos, de esta manera se pueden especificar varias iteraciones al diseño en una fase previa sin tener que recurrir a la modificación continua de código.
- Por otro lado, el uso de las guías de estilo permite realizar cambios en el personal sin la preocupación de dedicar tiempo al estudio de la plataforma, ya que, estará todo especificado en la guía de estilos.

7.4.4.1. Elementos generales.

Como hemos comentado anteriormente vamos a hacer uso de *Angular Material* para toda la parte de diseño de EasyRent ya que nos proporciona una serie de elementos con una alta calidad en cuanto a diseño, además de una gran facilidad a la hora de implementar sus componentes. A continuación, podemos observar dos elementos que aparecen considerablemente en nuestro diseño de interfaces, por un lado, el uso de tablas con paginación para mostrar la información sobre productos, reservas, etc. Por otro lado, el uso de una barra superior que sea responsive implementando el famoso menú de hamburguesa.

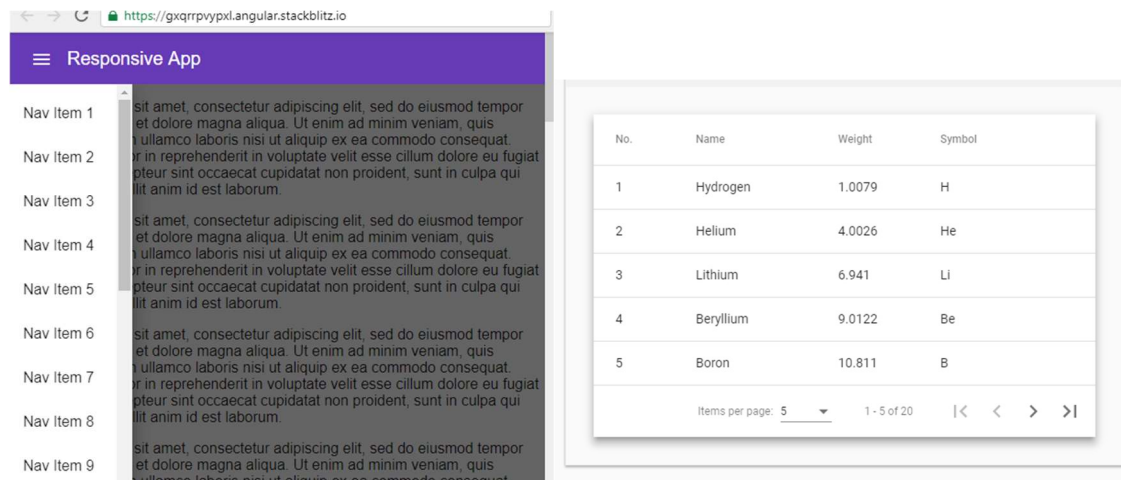


Figura 30. Menú responsive y tabla paginada. Fuente: <https://material.angular.io/components/>

7.4.4.2. Paleta de colores.

Como hemos comentado anteriormente vamos a emplear los componentes que nos suministra *Angular Material* únicamente añadiendo nuestro contenido y haciendo las modificaciones en cuanto a color para adaptarlo a un estilo concreto. Por esta razón es necesario hacer uso de una paleta de colores donde especificar los colores que se van a emplear por toda la interfaz.

Para ello haciendo uso de un conocido perfil de Instagram haremos uso de la siguiente plantilla de colores:

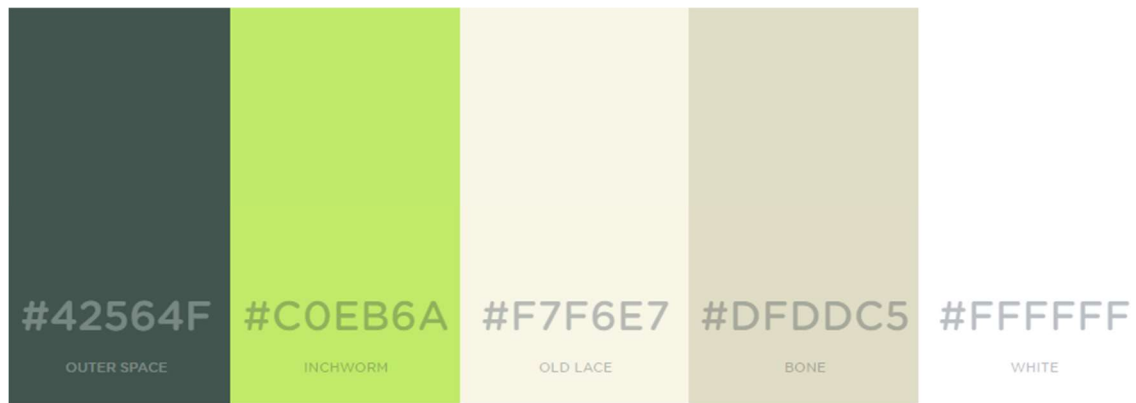


Figura 31. Paleta de colores EasyRent. Fuente:
https://www.instagram.com/p/Btiw2ihjy2/?utm_source=ig_share_sheet&igshid=h10gnplnnc84

Haciendo uso de la paleta de colores anterior podemos realizar los primeros diseños para la imagen corporativa, así como los logos que se van a emplear en las interfaces anteriormente descritas.

A continuación, podemos observar por un lado la imagen corporativa y por otro el diseño final del logotipo de la plataforma:



Figura 32: Imagen corporativa. EasyRent

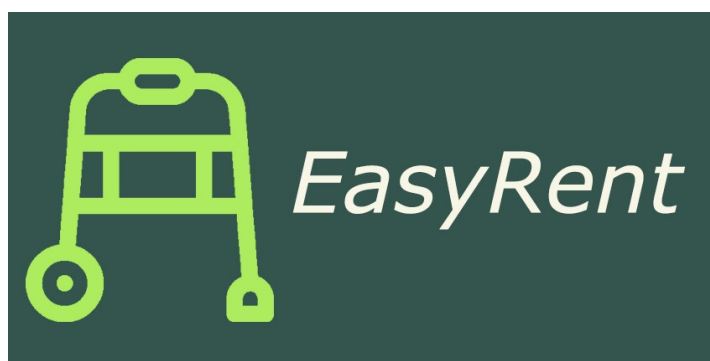


Figura 33: Logotipo de la plataforma

8. Implementación

A lo largo de este apartado vamos a comentar todos los aspectos relativos a la implementación de la propuesta de este proyecto, concretamente se van a abordar aspectos como el entorno de trabajo, metodología empleada e incluso los problemas que hayan podido surgir durante el desarrollo de éste. Como el proyecto se encuentra alojado en Git puede consultarse a través del enlace: <https://github.com/carlosGG15/TFG>

8.1. Entorno de desarrollo.

En primer lugar, comentaremos brevemente el entorno de desarrollo empleado, cabe destacar que todo el desarrollo se ha realizado con un sistema operativo Ubuntu 18.04:

- **WebStorm by Jet Brains:** tanto para la parte del servidor como la del cliente he empleado este IDE de desarrollo bajo la licencia de estudiantes, escogí esta opción debido a que me es familiar, ya que, actualmente en mi puesto de trabajo hago uso de ella y porque dispone de funcionalidades muy útiles como por ejemplo la conexión con la base de datos del sistema permitiendo la posibilidad de realizar operaciones sobre ella.[18]
- **MySQL Workbench:** en este caso hice uso de esta herramienta para la creación y la gestión de la base de datos ya que ofrece muchas posibilidades, ya que, permite la creación de bases de datos a partir de un modelo entidad relación y potentes herramientas de exportación e importación. [19][21]
- **GitHub:** en este caso emplee esta herramienta principalmente para disponer de una copia de seguridad en remoto ante posibles complicaciones, además de todas las bondades que presenta el uso de esta herramienta para el control de versiones y el trabajo en equipo. [20].
- **Postman:** en este caso hice uso de esta herramienta para comprobar el funcionamiento de las rutas y peticiones que iba desarrollando. [21]
- **Trello:** para planificar los Sprints de 15 días y establecer el estado de las tareas.

8.2. Metodología empleada.

En cuanto a la metodología empleada para el desarrollo de la plataforma se han seguido los principios de SCRUM como se comentaba en apartados anteriores estableciendo Sprints de 15 días donde se tienen que realizar una serie de tareas establecidas. Estos sprints realizados durante el desarrollo de la plataforma están sujetos a posibles cambios o modificaciones para adaptarse a los tiempos o las funcionalidades deseadas. A continuación, podemos observar una imagen de uno de los últimos sprints realizados durante el desarrollo de EasyRent.

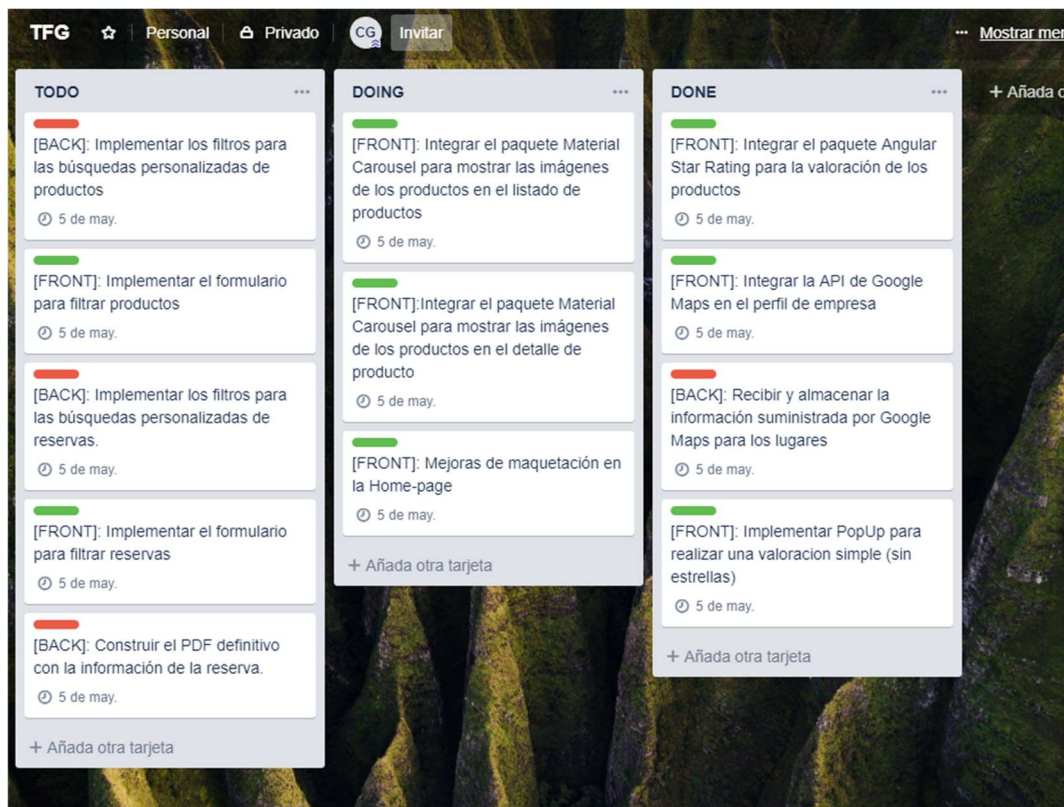


Figura 34. Sprint de EasyRent

En la imagen podemos observar que las tareas se clasifican en 3 estados en función de lo avanzadas que estén (TODO, DOING, DONE), también podemos observar que en el momento de realizar la fotografía se trata de un sprint activo, ya que, hay tareas en estado TODO.

Cabe destacar que la especificación de las tareas y el uso de sprints está muy relacionado con el uso de Git, ya que, en el momento que una de las tareas anteriores pase al estado DONE debe subirse a remoto para indicar que está realizada al 100% siguiendo la notación de las tareas indicando en el mensaje del commit la etiqueta y una descripción de las tareas realizadas. Un ejemplo de un commit sería el siguiente:

[FRONT]: Añade el mapa con el marcador de la dirección en el perfil de empresa.

Por último, en cuanto a la metodología empleada para el desarrollo de EasyRent me gustaría destacar que tanto el front-end como el back-end se han ido desarrollando de manera paralela para agilizar el proceso, es decir, la rutas de la API se iban implementando en el momento que eran necesarias para completar el funcionamiento de las vistas del front-end.

8.3. Implementación del back-end.

Como hemos comentado anteriormente el back-end de EasyRent está compuesto a grandes rasgos por una base de datos y de una API que permite comunicar el cliente con esos datos.

En primer lugar, gracias a la potente herramienta de generación de scripts que proporciona MySQL Workbench a partir de un modelo entidad relación podemos generar todo el código SQL necesario para la creación de la base de datos relacional.

Una vez creada la base de datos el siguiente paso fue la creación de la API en este caso el entorno de desarrollo Express nos proporciona un paquete muy interesante que nos permite crear la estructura básica de una aplicación en nuestro caso una API. Este paquete es *express-generator* que mediante el comando *express {nombreApp}* se genera el esqueleto básico.

A partir de ahí se han ido creando los servicios y las rutas atendiendo al patrón MVC siendo estos directorios los más importantes para el proyecto:

- **Routes:** Donde cada entidad va a tener su propio fichero con sus rutas, por ejemplo, todas las rutas relacionadas con una reserva estarán en un mismo archivo.
- **Controllers:** Donde cada entidad tendrá su propio archivo con toda la lógica de negocio.
- **Models:** En este caso con la ayuda del ORM Sequelize cada una de las tablas de la base de datos tendrá su fichero para poder mapearlas y poder convertirlas a Objetos y facilitar así las consultas a la base de datos.

Por otro lado, de la implementación de la API me gustaría destacar 3 archivos de vital importancia para el correcto funcionamiento de esta.

- **dbConnection:** este fichero contendrá toda la información relativa a la conexión con la base de datos mencionada anteriormente, así como los nombres de usuario, contraseña y el nombre del schema de base de datos.

- **dbModel:** este fichero es necesario para poder representar las relaciones entre las diferentes entidades de la base de datos entre los objetos mapeados con el ORM Sequelize.
- **enviroment.env:** en este caso este fichero es el encargado de albergar las variables de entorno que deben ser ocultadas, estas variables pueden ser, las claves de cifrado, el nombre de la base de datos, etc.

Cabe destacar que por seguridad y mediante el uso del fichero `.gitignore` estos archivos no se suben a remoto y el git no los tiene en cuenta a la hora de hacer un commit, ya que, por ejemplo si deseáramos desplegar la aplicación en un entorno de producción posiblemente haya que realizar modificaciones en estos archivos.

Por último, en cuanto a la implementación del back-end de EasyRent me gustaría destacar que se ha planteado el tema de los filtrados tanto de las reservas como de los productos de tal forma que sea sencillo de escalar, es decir, que sea muy sencillo añadir más campos por los cuales se pueda filtrar la información, esto lo podemos ver en más detalle en en [Apendice 1](#)

8.4. Implementación del front-end.

Como hemos comentado en secciones anteriores el front-end se ha implementado bajo uno de los frameworks más demandados del mercado, *Angular* en su versión 7, juntamente con *Angular material* para la maquetación y la reutilización de componentes, dejando las mejoras y aspectos de diseño a CSS puro.

Me gustaría destacar que al igual que *express* dispone de una línea de comandos para la generación del esqueleto de las aplicaciones angular dispone de *angular-CLI* que además de generar de manera automática la estructura básica de una aplicación, permite también la generación tanto de componentes, servicios, directivas, etc. Añadiendo las dependencias al archivo *app.modules.ts* lo que facilita en gran medida el desarrollo de aplicaciones en *Angular*.

En cuanto a la estructura de directorios del proyecto destinado al front-end me gustaría destacar las siguientes debido a su importancia en el correcto funcionamiento del proyecto:

- **App>>components:** Es la carpeta con más contenido, ya que, contendrá todos los elementos que se muestren en la pantalla, únicamente habrá que referenciar al componente con su correspondiente etiqueta html personalizada para poder emplearlo las veces que deseemos, este factor es una de las mayores ventajas de *Angular* en cuanto a la reutilización de código. Cada componente a su vez está formado

principalmente por dos archivos uno en TypeScript donde se situará toda la lógica asociada al componente y un HTML con la vista del componente.

- **App>>services:** Este directorio es muy importante ya que dispone de los archivos encargados de la obtención de datos desde un servicio externo a la aplicación Angular, en este caso son los archivos encargados de hacer las llamadas a nuestra API, cabe destacar que por temas de arquitectura que tabla/entidad de la base de datos tendrá su servicio asociado.
- **Enviroments:** En este caso al igual que en el back-end en este directorio se sitúan los archivos que van a almacenar las variables de entorno.

Además de estos directorios existen una serie de archivos que cobran gran importancia en el desarrollo y funcionamiento del proyecto:

- **App-routing.module.ts:** Este archivo es de vital importancia, ya que, es el encargado de establecer las rutas de la aplicación, es decir, a partir de un path o url es capaz de redirigir al componente asociado a ese path, sin la necesidad de actualizar la página.
- **App.module.ts:** Este fichero es el más importante del proyecto, ya que, es donde se almacenan todo tipo de demencias y referencias a los paquetes que se van a emplear dentro de la aplicación.

Por último, me gustaría comentar una de las muchas posibilidades de reutilización de código que ofrece *Angular* a sus desarrolladores, en este caso es el uso de las directivas personalizadas. Esta característica nos permite definir comportamiento específico a cualquier elemento del DOM. En el caso de EasyRent he empleado una directiva personalizada para integrar Google Maps Autocomplete. De esta manera podemos situar un input con esta funcionalidad en cualquier parte de nuestra aplicación emprendo el atributo *appGooglePlaces* para la etiqueta *input*.

A continuación, podemos ver con más detalle el funcionamiento de la directiva personalizada *appGooglePlaces*:

```

1 declare const google: any;
2 import { Output, Directive, ElementRef, EventEmitter, OnInit } from '@angular/core';
3
4 @Directive({
5   selector: '[appGooglePlaces]'
6 })
7 export class GooglePlacesDirective implements OnInit {
8   @Output() onSelect: EventEmitter<any> = new EventEmitter();
9   private element: HTMLInputElement;
10
11   // in the constructor and can be made readonly
12
13   ngOnInit(): void {
14     const autocomplete = new google.maps.places.Autocomplete(this.element);
15
16     google.maps.event.addListener(autocomplete, 'place_changed', () => {
17       this.onSelect.emit(this.getFormattedAddress(autocomplete.getPlace()));
18     });
19   }
20
21   getFormattedAddress(place) {
22     // @params: place - Google Autocomplete place object
23     // @returns: location_obj - An address object in human readable format
24     let location_obj: any = {};
25     console.log(place);
26     location_obj.IdLugar = place.place_id;
27     for (const i in place.address_components) {
28       const item = place.address_components[i];
29       location_obj.Dirección = place.formatted_address;
30       if (item.types.indexOf('locality') > -1) {
31         location_obj.locality = item.long_name;
32       } else if (item.types.indexOf('administrative_area_level_1') > -1) {
33         location_obj.admin_area_l1 = item.short_name;
34       } else if (item.types.indexOf('street_number') > -1) {
35         location_obj.street_number = item.short_name;
36       } else if (item.types.indexOf('route') > -1) {
37         location_obj.route = item.long_name;
38       } else if (item.types.indexOf('country') > -1) {
39         location_obj.country = item.long_name;
40       } else if (item.types.indexOf('postal_code') > -1) {
41         location_obj.postal_code = item.short_name;
42       }
43     }
44     location_obj.Latitud = place.geometry.location.lat();
45     location_obj.Longitud = place.geometry.location.lng();
46     return location_obj;
47   }
48 }

```

Figura 35. Google-places.directive.ts. Fuente propia

```

23 <mat-form-field class="inputElement" >
24   <input formControlName = "Dirección" appGooglePlaces (onSelect)="selectPlace($event)" type="text" matInput placeholder="Dirección">
25 </mat-form-field>

```

Figura 36. Ejemplo de input con la directiva. Fuente propia



Figura 37. Resultado de la directiva. Fuente propia

En el caso concreto de la directiva personalizada `appGooglePlaces` defines un evento que recupera el lugar introdujo por el usuario y lo formatea con los datos necesarios para nuestro caso concreto.

8.5. Problemas durante el desarrollo.

Como en cualquier proceso de desarrollo de software es inevitable que surjan complicaciones o problemas. Estos problemas pueden deberse a muchos factores, pero los más comunes son:

- Falta de tiempo.
- Mala planificación
- Desconocimiento de la tecnología que implica tiempo de aprendizaje.
- Otros.

Por tanto, EasyRent no iba ser menos y durante el desarrollo me encontré ante algunos problemas de los cuales no todos pudieron ser resueltos, haciendo que alguna funcionalidad se viera afectada o simplemente se descartara.

El primer de los problemas que he encontrado es el desconocimiento para realizar búsquedas por proximidad, junto al tiempo limitado para realizar la implementación, por este motivo esta funcionalidad se ha descartado.

Otro problema encontrado durante el desarrollo debido nuevamente a falta de conocimiento con CSS y con el tiempo limitado para realizar un diseño muy llamativo y atractivo, ya que, durante el desarrollo he considerado más importante y he priorizado realizar las funcionalidades troncales del sistema de manera correcta, así como el uso de una buena arquitectura frente el diseño.

Por último me gustaría destacar un problema que surgió durante el desarrollo del front-end debido al funcionamiento de Angular y la metodología SPA (single page application), ya que, por temas de arquitectura el menú superior solo se renderiza una vez ya que siempre va a ser el mismo componente, por tanto, el problema apareció en el momento que el usuario hacía login, ya que, debía modificar los enlaces en función del tipo de usuario sin la posibilidad de actualizar la página para cargar un componente diferente.

Para solucionar este detalle fue necesario realizar un “servicio” de autenticación en el cliente que notifica al componente del menú superior si el usuario ha iniciado la sesión y el tipo de

usuario que es para así poder adaptar dicho menú. Para lograr esto fue clave entender el funcionamiento de los *observables* y los *eventos* de los que dispone Angular.

9. Resultado final.

A lo largo de esta sección se va a dejar constancia del trabajo realizado de manera visual, ya que, vamos a mostrar el resultado final de la plataforma tanto a nivel funcional, como a nivel estético. Cabe destacar que este resultado no se corresponde con un producto finalizado, debido a la naturaleza académica del proyecto, en apartados posteriores concretaremos los aspectos necesarios para cerrar el proyecto de manera “definitiva”.

En primer lugar, me gustaría destacar que el diseño de todas las interfaces de EasyRent se han desarrollado de manera *responsive*, es decir, que se puedan visualizar correctamente en cualquier dispositivo sin importar el tamaño de éste. Esto se puede observar claramente en la *Home-page* y el menú de navegación de la plataforma. También se puede apreciar el uso de la paleta de colores descrita en apartados anteriores.

9.1. Home-Page.

A continuación, podemos observar la home-page de EasyRent en dos casuísticas diferentes. En todos los casos disponemos de una parte común para realizar una búsqueda de los productos atendiendo a los criterios de búsqueda por nombre o eligiendo una de las empresas dadas de alta en la plataforma.

Los casos anteriormente comentados están basados principalmente en función del tamaño de la pantalla de visualización y de si un usuario ha iniciado sesión en el sistema.

En la primera imagen podemos observar el aspecto que presenta la página principal, con el menú superior con las opciones de iniciar sesión o de registrarse, así como la zona para realizar la búsqueda anteriormente mencionada y de un footer informativo sobre la plataforma.

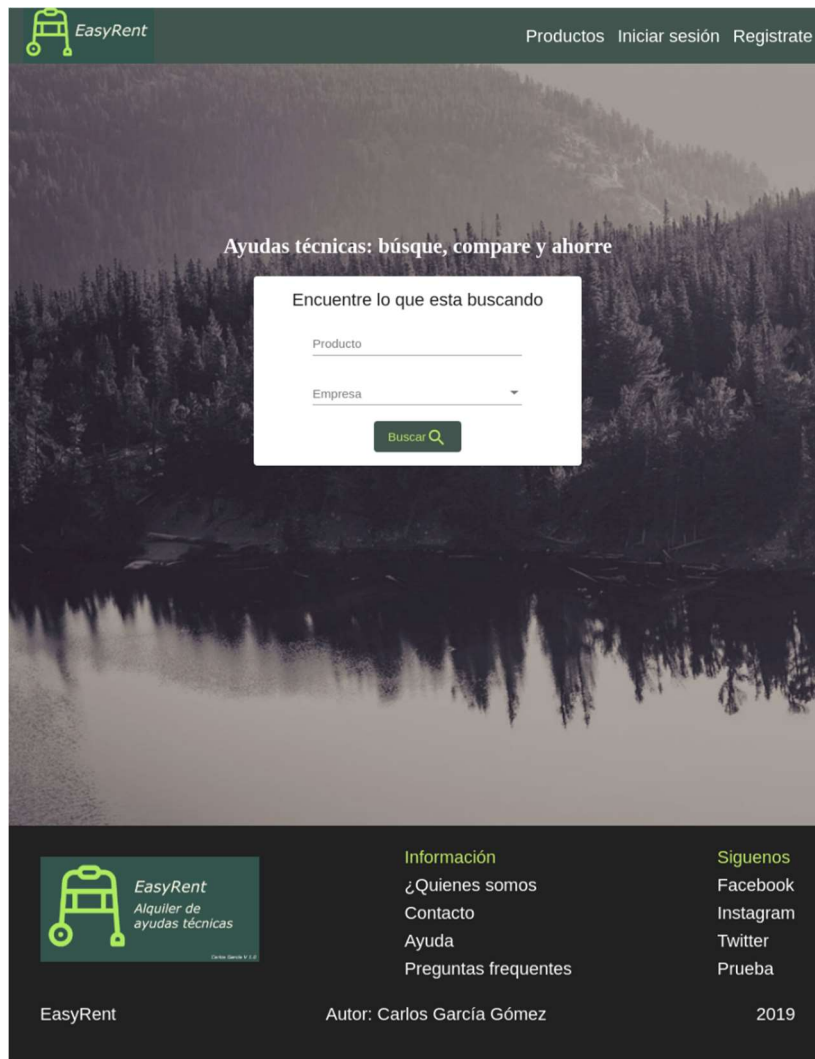


Figura 38 Home-page EasyRent. Fuente propia

En las siguientes dos imágenes podemos apreciar el segundo caso de para la visualización de la plataforma en un dispositivo de menor tamaño y con una sesión iniciada dentro de la plataforma. Cabe destacar que el menú superior se convierte en un menú de *Hamburguesa* ocultando las opciones de menú. Para desplegar en este caso el menú y poder visualizar las opciones debemos hacer clic en el icono de menú (tres rallas).

Una vez desplegado el menú podemos observar que al haber iniciado sesión en la plataforma estas opciones cambian respecto a la *Figura 38* Home-page EasyRent. Fuente propia, ya que en este caso disponemos de cierta información respecto al usuario que ha iniciado sesión dentro de EasyRent.



Figura 39. Home-page móvil.

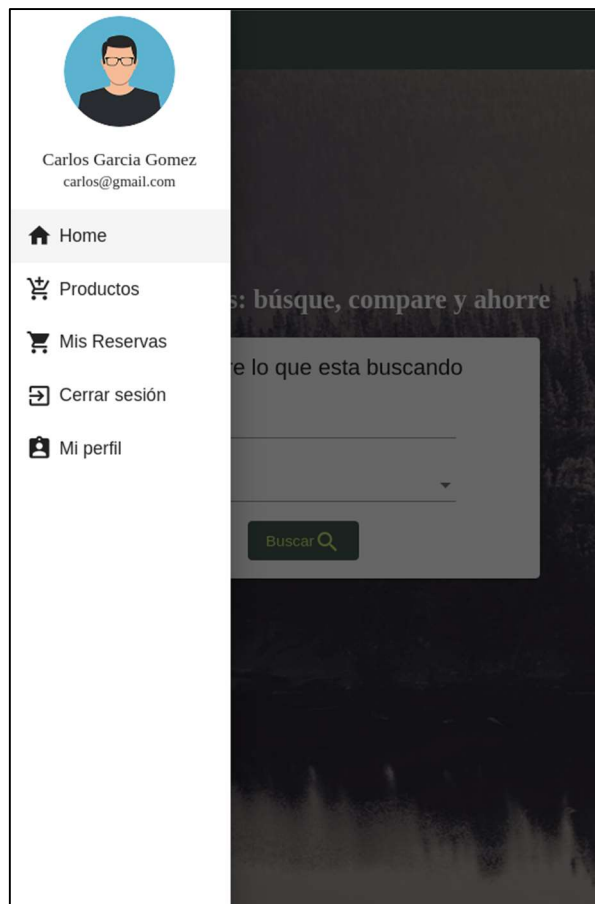


Figura 40. Home-page móvil menú desplegado

9.2. Login/Registro.

Como hemos mencionado anteriormente los usuarios de EasyRent tendrán la posibilidad de registrarse en el sistema para desbloquear la mayor parte de las funcionalidades del sistema, para ello disponen de una interfaz propia para ello. Esta interfaz además de disponer los elementos comunes como son el menú de navegación y el footer informativo mencionados anteriormente, cuenta con una zona de pestañas que nos permiten selección si deseamos iniciar sesión o darnos de alta en la plataforma.

En el caso de iniciar sesión el usuario únicamente deberá introducir su email y contraseña, en el caso de que estos dos campos sean correctos accederá a la plataforma y el sistema le redirigirá al listado de producto para que pueda realizar una búsqueda.

EasyRent

Productos Iniciar sesión Registrate

EasyRent

Iniciar sesión Registrate Registro Empresa

Email *

carlos@gmail.com

Contraseña *

Iniciar sesion

EasyRent Alquiler de ayudas técnicas

Información

¿Quienes somos

Contacto

Ayuda

Preguntas frecuentes

Siguenos

Facebook

Instagram

Twitter

Prueba

EasyRent

Autor: Carlos García Gómez

2019

Figura 41. Login EasyRent. Fuente propia.

Por otro lado, si un nuevo usuario se dirige a la pestaña de registro únicamente deberá rellenar los campos que le solicita la interfaz, atendiendo a las validaciones como los campos requeridos, la seguridad de la contraseña, así como que el email introducido esté disponible en el sistema.

Si el registro se realiza de manera correcta el sistema realizará el login de manera automática comportándose como hemos mencionado anteriormente.


Figura 42. Registro EasyRent. Fuente propia.

9.3. Perfil de usuario(empresa/Cliente).


Como hemos mencionado en el apartado anterior un usuario de la plataforma puede iniciar sesión en el sistema para desbloquear todas las funcionalidades de EasyRent. Una de ellas es la posibilidad de visualizar y modificar la información relativa al usuario, para ello es necesario hacer uso de la interfaz de ver perfil.

Cabe destacar que esta interfaz es común tanto para un usuario de tipo cliente como para uno de tipo empresa, salvo que, en el caso de tratarse de una empresa permite modificar más información (descripción, lugar...).

A continuación, podemos observar la interfaz para un usuario de tipo empresa:


EasyRent

[Productos](#)
[Cerrar sesión](#)
[Empresa Mod 34](#)



Nombre

Empresa Mod 34

Email

empresa@gmail.com

Password

Descripción de la empresa


Probando


Dirección

Alicante (Alacant), Alicante, España

Cambiar

Guardar





EasyRent

Alquiler de ayudas técnicas

Información

¿Quiénes somos

Contacto

Ayuda

Preguntas frecuentes

Síguenos

Facebook

Instagram

Twitter

Prueba

EasyRent

Autor: Carlos García Gómez

2019

Figura 43. Perfil empresa. Fuente propia

Por último, en cuanto a esta interfaz me gustaría destacar la integración con Google Maps, para colocar un marcador en el mapa en función del lugar que seleccione el usuario en el campo de Dirección mediante la ayuda de *Google places Autocomplete*.

9.4. Listado productos.

Para cumplir con los requerimientos establecidos en apartados anteriores, se ha desarrollado una interfaz para que los usuarios puedan visualizar la oferta de productos disponible en EasyRent.


Cabe destacar que como todas las interfaces de EasyRent dispone de un menú de navegación variable en función del tipo de usuario que haya iniciado la sesión, así como un footer informativo de la plataforma.





En primer lugar, en la parte superior disponemos de una zona para aplicar filtros para una búsqueda, estos filtros permiten buscar por uno o por una combinación de los campos siguientes:

- Nombre del producto.
- Empresa que lo distribuye.
- Tipo de producto (manual, con motorización).
- Precio que sea menor al indicado.


Por otro lado, el usuario dispone de una zona de visualización de 10 productos paginados, donde podemos ver de manera rápida y visual las imágenes del producto, el nombre, la descripción, el precio y la valoración media a razón del número de valoraciones que tenga el producto.

Por último, cabe destacar que para acceder a la información detalla de cada uno de los productos del listado basta con pulsar encima del contenedor del producto o en el botón destinado para ello. Entraremos en más detalle en cuanto a la visualización de un producto de manera individual en el siguiente apartado.


EasyRent
Productos
Iniciar sesión
Regístrate

	Andador Hi Lowi Andador con asiento regulable en altura, 4 diferentes opciones. Puede ser utilizado tanto en interiores como en exteriores. Fabricado en aluminio.	★★ ★ 123/ Día <input type="button" value="Ver detalle"/>
	Air-On Zero El andador Air-On Zero de Forta está fabricado en hierro y termoplástico. Con empuñaduras ergonómicas y regulables en altura, permiten un agarre cómodo y confortable. Fácil de manejar y económico.	★★ ★★ ★ 12/ Día <input type="button" value="Ver detalle"/>
	Taurus 2 Taurus 2 es ligero, con una estética y diseño modernos además de una excelente maniobrabilidad.	★★ ★★ 75/ Día <input type="button" value="Ver detalle"/>
	Carbon Ultralight Andador rollator de carbono Carbon Ultralight fabricado en fibra de carbono y en tres colores a elegir.	★★ ★ 90/ Día <input type="button" value="Ver detalle"/>

Items per page: 10
 1 - 4 of 4
 <
 >



EasyRent
 Alquiler de
 ayudas técnicas
(Carlos García M. S.L.)

Información
 ¿Quiénes somos
 Contacto
 Ayuda
 Preguntas frecuentes

Síguenos
 Facebook
 Instagram
 Twitter
 Prueba

EasyRent
 Autor: Carlos García Gómez
 2019

Figura 44. Listado productos. Fuente propia


9.5. Detalle producto.

Como hemos visto en el apartado anterior los usuarios tienen a su disposición un listado de productos, que pueden filtrarlos mediante los criterios de búsqueda que deseen. Si un usuario está interesado en uno de estos productos y decide visualizar más información acerca del producto hace uso de la interfaz de ver detalle de producto.


A nivel funcional esta interfaz es la más compleja, ya que, engloba 3 funcionalidades básicas y necesarias para la plataforma estas funcionalidades son ver detalles del producto, realizar una reserva y realizar una valoración.

Este interfaz como todas las anteriores dispone de dos zonas comunes, el menú de navegación y el footer con información acerca del EasyRent.


En cuanto a la visualización de los detalles del producto, podemos visualizar la descripción del producto, las imágenes de éste mediante un carrousel de fotos, el precio por día de alquilarlo y la empresa a la que pertenece.



EasyRent
Productos
Mis reservas
Cerrar sesión
Maria

Andador Hi Lowi :Empresa Mod 34



Andador rolator Hi Low con asiento regulable en altura para poder adaptarse a ti, hay 4 diferentes opciones que van de 52 a 60 cm.. Está diseñado para poder ser utilizado tanto en interiores como en exteriores. Fabricado en aluminio y con soporte para bastón.


Desde: 

Hasta: 

Precio: 123 €/día

[Reservar](#)

Valoraciones [Valorar](#)




Cumple justito 2019-05-12

Maria: maria@gmail.com

★★★★

Cumple justito, porque al ser alquilado no estaba en buen estado. Ha servido para que mi madre decida comprar uno parecido




Perfecto 2019-05-12

Pepe: pepe@gmail.com

★★★★★

El producto tiene una relación calidad precio espectacular.




Fantástico 2019-05-12

Carlos Garcia Gomez: carlos@gmail.com

★★★★★

El producto cumple con su propósito. A mi me ha ayudado muchísimo para poder volver a caminar



EasyRent
Alquiler de ayudas técnicas

Información

- ¿Quiénes somos
- Contacto
- Ayuda
- Preguntas frecuentes

Síguenos

- Facebook
- Instagram
- Twitter
- Prueba

EasyRent

Autor: Carlos García Gómez

2019

Figura 45. Detalle producto. Fuente propia.

Por otro lado, si el usuario quiere realizar una reserva únicamente tendrá que introducir la fecha de inicio y la fecha final para el alquiler y el sistema le calculará automáticamente el precio de la

reserva. Si el usuario está conforme con este precio pulsará en reserva y la reserva se hará efectiva mostrando un dialogo de confirmación.

Por último, la interfaz de detalle de producto dispone de un listado de valoraciones con la información del usuario que la ha realizado, así como la fecha en la que se ha publicado la valoración y el contenido de esta. Si un usuario desea realizar una valoración deberá pulsar el botón de valorar y le aparecerá un dialogo que le permitirá introducir un título, su opinión y el número de estrellas a modo de calificativo, cabe destacar en este caso el uso del paquete *angular-star-rating* para dar la posibilidad al usuario de valorar el producto mediante el uso de estrellas.

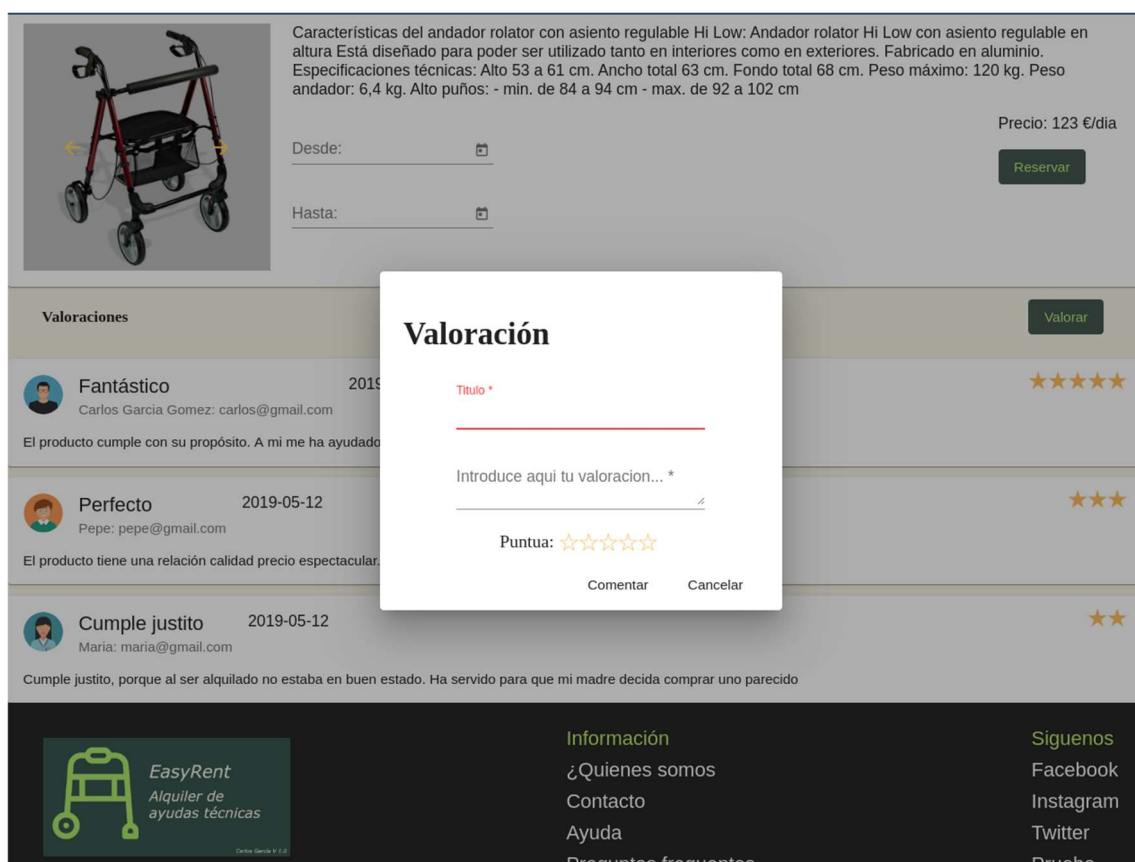



Figura 46. Dialogo para añadir valoracion. Fuente propia

9.6. Mis Reservas.

















Como hemos mencionado en apartados anteriores un usuario registrado en EasyRent podrá realizar reservas dentro de la plataforma, en el apartado anterior se especificaba el proceso que debía seguir el usuario para realizar la reserva. Una vez un usuario ha realizado una reserva podrá visualizar la información de sus reservas en el apartado de reservas del sistema.

Este apartado está compuesto como todas las interfaces de la plataforma de unas zonas comunes como son el menú de navegación y del footer informativo. Además de esto la interfaz presenta de una tabla con la información básica de cada una de las reservas que ha realizado el usuario, además de una zona que permite filtrar la información de la tabla. Esta tabla dispone de una paginación que limita las filas de la tabla a 10 elementos, evitando así una página un *scroll* excesivo.


En cuanto a la tabla me gustaría destacar la columna de acciones, cuyo contenido son dos iconos que permiten realizar funcionalidades muy diferentes. Por un lado, el ojo permite acceder a una página donde se muestran más detalles de la reserva y por otro el icono de la impresora brinda la posibilidad al usuario de generar un PDF con toda la información de la reserva, en el siguiente apartado entraremos en más detalle en estos aspectos.


EasyRent
Productos
Mis reservas
Cerrar sesión
Carlos García Gomez

Desde:
Hasta:
Producto
Precio
Buscar

Id	Desde	Hasta	Producto	Precio	Acciones
1	2019-04-26	2019-04-29	Producto 1	96 €	 
2	2019-05-02	2019-05-05	Producto 1	96 €	 
3	2019-05-02	2019-05-06	Producto 1	120 €	 
4	2019-05-02	2019-05-05	Producto 1	96 €	 
5	2019-05-27	2019-05-29	Producto 1	72 €	 
6	2019-05-14	2019-05-16	Producto 1	72 €	 
7	2019-05-28	2019-05-30	Producto 1	72 €	 
8	2019-05-09	2019-05-12	Andador Hi Lowi	492 €	 

Items per page: 10
1 - 8 of 8
<
>



EasyRent
Alquiler de ayudas técnicas

Información
¿Quiénes somos
Contacto
Ayuda
Preguntas frecuentes

Síguenos
Facebook
Instagram
Twitter
Prueba


EasyRent
Autor: Carlos García Gómez
2019

Figura 47. Mis reservas. Fuente propia.

9.7. Detalle reserva.


Los usuarios de la plataforma tienen la posibilidad de ampliar la información de las reservas de la tabla del apartado anterior pulsando en la fila o en el icono del ojo. Esta interfaz de detalle de reserva muestra toda la información del producto reservado, así como la información propia de la reserva. A continuación, en la imagen siguiente podemos visualizar una reserva realizada por un usuario.

Cabe destacar que la implementación de esta interfaz fue relativamente sencilla, debido a que, comparte muchos elementos de la interfaz de visualizar el detalle de un producto, ya que, la interfaz es prácticamente la misma eliminado la parte de las valoraciones del producto.

 EasyRent

Productos Mis reservas Cerrar sesión Maria

Reserva: Andador Hi Lowi



Andador Hi Lowi


Andador rolator Hi Low con asiento regulable en altura para poder adaptarse a ti, hay 4 diferentes opciones que van de 52 a 60 cm.. Está diseñado para poder ser utilizado tanto en interiores como en exteriores. Fabricado en aluminio y con soporte para bastón.

Precio por día: 123

Desde: 5/12/2019

Hasta: 5/16/2019

Precio: 615 €

 EasyRent
Alquiler de ayudas técnicas
Carlos García V. 2.01

Información

¿Quiénes somos

Contacto

Ayuda

Preguntas frecuentes

Síguenos

Facebook

Instagram

Twitter

Prueba

EasyRent

Autor: Carlos García Gómez

2019

Figura 48. Detalle reserva. Fuente propia

Por otro lado, podemos observar que el usuario tiene a su disposición un botón que le permite imprimir la reserva, si el usuario hace clic en el botón el sistema generara un PDF con toda la información de la reserva que se podrá visualizar en una nueva pestaña del navegador, permitiendo así que el usuario pueda descargárselo y llevarlo posteriormente al punto de entrega/recogida.

A continuación, podemos observar un ejemplo de un PDF generado por la plataforma para la reserva anterior, este documento muestra una imagen del producto, así como los datos propios de la reserva como son las fechas y el precio total.



Reserva numero: 8

Andador Hi Lowi



La reserva para el producto *Andador Hi Lowi* de la empresa *Empresa Mod 34* se ha realizado correctamente.

Desde: 2019-05-09 hasta: 2019-05-12 por 492 €

Gracias por confiar en nosotros.

Para más información visiste www.easyRent.com

Figura 49. Reserva en PDF. Fuente propia

10. Conclusiones y trabajo futuro.

10.1. Conclusiones.

A partir de todos los apartados anteriores podemos afirmar que EasyRent es una plataforma funcional para poner en contacto a las personas que necesitan hacer uso de alguna ayuda técnica con las empresas que las distribuyen, por tanto, podemos afirmar que se ha cumplido todos los objetivos y requerimientos principales abordados en este proyecto.

Cabe destacar que desarrollar un proyecto de estas características nos permite abordar todos los aspectos del desarrollo software haciendo que cambie mi manera de ver un proyecto y ser capaz de detectar e identificar los conflictos y las dificultades que pueden surgir durante el desarrollo de éste, debido a mala planificación, falta de conocimiento, etc.

A nivel personal estoy bastante satisfecho con el resultado del proyecto, ya que, he sido capaz de compaginar las últimas asignaturas del grado, con un trabajo, además de una de mis pasiones como es jugar en un equipo de balonmano. Sin embargo, siendo realista el estado actual de este proyecto dista en cierta medida de uno profesional, debido su carácter académico, así como por la falta de tiempo para desarrollarlo.

A pesar de lo anterior, EasyRent surge de una necesidad del día a día y que está más frecuente en la sociedad de lo que la mayoría pensamos, por esto creo que hoy en día sigue sin haber una solución tecnológica que solucione esta problemática y que con algo de financiación y mayor tiempo para el desarrollo puede llegar a ser un caso de éxito.

Por último, me gustaría destacar que gracias a la realización del TFG he podido poner en práctica y sobre todo mejorar muchas de las habilidades obtenidas durante los años de estudio en esta universidad y poder afirmar que estoy preparado para enfrentarme al mundo laboral de manera efectiva y solvente.

10.2. Trabajo futuro.

Como se ha podido observar en el apartado de *Resultados* EasyRent no es un producto totalmente acabado, por lo que sería necesario realizar trabajo en un futuro para que pudiera llegar a comercializarse de manera oficial.

El trabajo futuro para lograr el propósito anterior se podría desglosar en los siguientes grupos de conocimiento/trabajo:

En primer lugar, sería necesario realizar una mejora bastante considerable en cuanto a diseño para que el producto de la sensación de acabado y para que pueda ser un factor diferencial a la hora de buscar cualquier tipo de ayuda técnica colocando a EasyRent en la vanguardia en cuanto a diseño de interfaces.

En segundo gran bloque de trabajo consiste en ampliar las funcionalidades ofrecidas por la plataforma, las que a mi juicio serían las más importantes para lograr que el producto pudiera comercializarse sería las siguientes:

- Incluir un registro social mediante el uso de OAuth para poder iniciar sesión usando las cuentas de compañías tan potentes como pueden ser Google o Facebook para dotar a EasyRent de cierto prestigio para que el usuario que haga uso de la plataforma se sienta arropado por organismos tan importantes.
- Añadir una pasarela de pago para que las reservas lleguen a efectuarse de manera más efectivas, y para que el sistema de comisiones cobre sentido realizando los pagos de manera automática, cerrando así el círculo del proceso de reservar una ayuda técnica.
- Cumplir el requerimiento inicial de poder filtrar los resultados de una búsqueda por proximidad, de esta manera el usuario podrá encontrar lo que busca lo más cerca suya posible.

Por último, en cuanto al trabajo en un futuro me gustaría hacer una clasificación con las labores de despliegue, mantenimiento, seguridad y copias de seguridad del sistema una vez se haya realizado el despliegue en un servidor de producción, ya que, es un aspecto muy importante, debido a que puede suponer algún que otro problema o puede implicar algún tipo de modificación en el código realizado hasta la fecha.

Referencias

A continuación, se establece un listado de todas las referencias de las que he hecho uso para el desarrollo de esta memoria:

1. ¿Qué es una campaña SEM?:
 - <https://neoattack.com/que-es-una-campana-sem/>
2. ¿Qué entendemos por SMM?
 - <https://www.websa100.com/blog/social-media-marketing-smm/>
3. ¿Qué es una campaña Mailing?
 - <https://www.mailify.com/es/email-marketing>
4. ¿Qué es un motor de búsqueda o buscador?
 - https://es.wikipedia.org/wiki/Motor_de_b%C3%BAsqueda
5. Como los buscadores han evolucionado a los actuales Metabuscadores:
 - <https://es.wikipedia.org/wiki/Metabuscador>
6. Prestaciones específicas de la Comunidad Valencia:
http://www.san.gva.es/web/dgfps/prestaciones-especificas-cv;jsessionid=B4E4CADFDB55A538CB46649977C47264.appli7_node2
7. CMS: Sistema de gestión de contenidos
8. BackOffice: Parte de gestión y administración de una herramienta web.
9. Gonzalo Gabino Diestro: *Qué son las metodologías ágiles y su aplicación en el mercado actual*. Universidad de Valencia.
 - https://www.uv.es/capgeminium/documents/Capgeminium_Charta_Agile_UV.pdf
10. ExtremeProgramig: Metodología ágil. Disponible en:
https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema
11. Toggl. Herramienta para contabilizar el tiempo dedicado a una tarea. Disponible en:
<https://toggl.com/>
12. Jira: Herramienta de Atlassian dedicada a la gestión de proyectos. Disponible en:
<https://es.atlassian.com/software/jira>
13. Microsoft Project: Herramienta especializada en la gestión de proyectos por parte de Microsoft. Disponible en: <https://products.office.com/es-es/project/project-and-portfolio-management-software>
14. Requisito funcional. https://es.wikipedia.org/wiki/Requisito_funcional

15. Requisito no funcional. https://es.wikipedia.org/wiki/Requisito_no_funcional
16. Arquitectura Cliente-Servidor: <https://es.wikipedia.org/wiki/Cliente-servidor>
17. Flat design: <https://www.departamentodeinternet.com/que-es-flat-design-diseno-plano/>
18. WebStorm by JetBrains: IDE de desarrollo especializado en JavaScript. Disponible en: <https://www.jetbrains.com/webstorm/>
19. MySQL Workbench: entorno de gestión y creación de bases de datos relacionales. Disponible en: <https://www.mysql.com/products/workbench/>
20. GitHub: herramienta para el control de versiones: Disponible en: <https://github.com/>
21. Postman. Herramienta para análisis de API REST. En forma de plugin para Chrome y aplicación. Disponible en: <https://www.getpostman.com/>

Apéndice I

A continuación, se detalla el proceso para obtener un resultado de búsqueda de productos mediante el uso de Sequelize de manera que sea escalable y que de una manera muy sencilla se puedan añadir nuevos criterios de búsqueda sin realizar modificaciones profundas en el código.

Para lograr el objetivo anterior se ha dividido la funcionalidad de buscar en dos métodos:

El primero de ellos es el que recibe la petición por parte del cliente cuyo cuerpo de la petición será un objeto JSON con los campos por los que el usuario desea filtrar la información y la página que solicita el usuario será la recibida como parámetro de la petición, como hemos comentado en la especificación de rutas en el apartado: *Diseño API*.

```
1 {  
2   "Nombre": "Nombre del producto",  
3   "IdUsuario": "Id de la empresa a la que pertenece",  
4   "Precio": 15  
5 }
```

Figura 50. Ejemplo del filtro recibido

Por otro lado, el tamaño de la página será fijo y se establece en la variable *limit* del método. Una vez tenemos la página solicitada y el tamaño de página hacemos uso del segundo método *buildQuery* que es el encargado de construir la cláusula *where* de la consulta SQL a partir del filtro recibido.

```
71  
72 /**  
73  * Metodo que construye la clausula where para hacer los filtrados  
74  * @param filtro  
75  */  
76 getQuery = function(filtro, callback) {  
77   console.log("LLEgo");  
78   let where = {};  
79   if(filtro.Nombre) {  
80     where.Nombre = {[Op.like]: '%'+filtro.Nombre+'%'};  
81   }  
82   if(filtro.IdUsuario){  
83     where.IdUsuario = filtro.IdUsuario;  
84   }  
85   if(filtro.Precio) {  
86     where.Precio = {[Op.lt]: filtro.Precio};  
87   }  
88   if(filtro.Tipo) {  
89     // TODO:  
90   }  
91   return callback(where);  
92 };
```

Figura 51. Método para generar la cláusula where. Fuente propia

El método de la imagen anterior simplemente comprueba que el filtro contiene un valor de la tabla por el que se puede filtrar y lo añade al objeto *where*, por tanto, en el caso que se desee añadir un nuevo criterio de búsqueda bastaría con añadir una nueva condición en este método, sin necesidad de modificar la consulta en sí.

Por último, hacemos uso de Sequelize empleando los parámetros generados antes de realizar la consulta y parseamos los resultados para cumplir con el formato comentado en apartados anteriores para la respuesta de una petición, cabe destacar que en el caso de que la consulta no obtenga resultados se construye una respuesta informando de lo sucedido.

```
36 ProductoController.findProductoByFiltro = function(req, res, callback) {
37   console.log(Op.substring);
38   const filtro = req.body;
39   let paginaActual = req.params.pag;
40   limit = 10;
41   let where = {};
42   getQuery(filtro, callback: function (result) {
43     console.log(result);
44     where = result;
45   });
46   Producto.findAndCountAll({ options: { where: where } }).then( onfulfilled: data => {
47     let numelems = data.count;
48     let paginasTotales = Math.ceil( (x: data.count / limit),
49     offset = limit * (paginaActual -1);
50     Producto.findAll( options: {
51       where: where,
52       limit: limit,
53       offset: offset,
54       include: [{
55         model: Imagen,
56         attributes: ['Data']
57       }],
58     }).then( onfulfilled: result => {
59       // Construyo el objeto de respuesta
60       console.log(result.length);
61       let productos = [];
62       for(let i = 0; i < result.length; i++) {
63         productos.push(result[i].dataValues);
64       }
65       res.json({success:true, message: 'Resultado búsqueda', data:{PaginaActual: paginaActual, count: numelems, productos}});
66     }).catch( onrejected: err => {
67       console.log(err);
68     })
69   })
70 };
71
```

Figura 52. Método general para obtener productos filtrados. Fuente propia